ErisX

FIX 4.4 Specification V3.13
Market Data and Order Entry

Please contact ErisX sales representatives or help desk personnel for more information on this documentation.

Contents

1 Change History

Date	Message(s) or Section	Description
2019-09-10	NEW T.M.E. FIX SPEC V2	This updated spec is provided for members to migrate to the new ErisX match engine coming in Q4 2019.
	V2.1	
2019-09-20	CancelReplace Overfill Protection	Overfill Protection is an optional attribute for order modification See
2019-09-20	Stop Order Time in force	TimeinForce FOK, IOC will be supported for stop limit orders.
2019-09-30	4.1 Sequence Number management	Updated to include note around Sequence reset messages sent by client applications.
2019-10-09	Overfill Protection	Removed incorrect note.
	V2.2	
2019-10-14	System Status and Security Status	Added detail around the system and security status messages for clients to understand the workflow when connecting and subscribing to data.
2019-10-21	V3.0	Includes Futures tags and messages.
	7.1 Order Input and Execution	Table updated.
	9.2.1.0 MarketDataIncremental	Tag 60 moved out of the MDIncGrp repeating group.
		tag 278 MDEntryID is not sent for statistics messages
		Duplicate tag 15 removed
	8.10 Mass Order Status Request	Updated with description of tag 912=Y
	V3.1	
2019-12-05	9.2.4 Security List Message	Added <EvntGrp> and associated tags for Futures.
		Added Tag 207 SecurityExchange
	9.2.5 Security Status Message	Added Tag 207 SecurityExchange
	9.2.7 Security Definition Message	Added Tag 207 SecurityExchange
	8.5 On Behalf of Routing	Updated description for on behalf of routing to refer to account and not user
2019-12-18	9.2.14 OrderReplace	Added Tag 59 = Time In Force
	V3.2	
2020-02-03	7.4 Top Of Book - Removed	Removed support for Top Of Book market Data. Users should refer to the WebSocket API for use of TopOfBook data.
2020-02-28	V3.3	Correction to spec: ClOrdID is only 40 characters
2020-03-02	V3.4	
	Supported Order Types	Added Post-Only as as Supported Order type in 3.1
	NewOrderSingle, ExecutionReport, OrderReplace	Added Tag 18 ExecInst 6 = Participate don't initiate
2020-03-20	MarketDataIncrementalRefresh	Added tag 346 to Ticker MarketDataIncrementalRefresh
	NewOrderSingle	Change Tag 40 Product to N - Not Required
	OrderReplace	Removed modify to IOC or FOK
	SecurityStatus 35=f	Tag 167 is only sent on Futures instruments
	Execution Report for OrderMassStatusRequest	Removed Commission group as this is currently not included.
	V3.6	
2020-04-23	SecurityList	Add tag 868 Event Time

	OrderReplace	Tag 59 changed to optional
	V3.7	
	Sequence Number Weekend Reset	Added new section describing weekend FIX sequence number reset logic and password expiration checks.
2020-06-03	V3.8	
	OrderCancelRequest, OrderCancelReject	Added Parties group when cancelling all orders on behalf of another FIX session.
	Cancel all Orders	A description of how to use the cancel all order functionality
	Execution Report (response sent to the OrderMassStatusRequest)	Added back CommissionData Component
	MarketDataIncrementalRefresh	Removed tag 110 as this is not supported
2020-07-09	V3.9	
	SecurityList	Add tag ProductCode(6005), tag SecurityGroup(1151), tag Cap (1199), tag Floor(1200) and tag LastFragment(893)
	ExecutionReport	Add AvailableBalanceData component New value in OrdRejReason(103), 100 = Instrument Closed New value in OrdRejReason(103), 23 = Quantity Above Account Asset Balance Limit
	Execution Report (response sent to the OrderMassStatusRequest)	Add AvailableBalanceData component
	OrderCancelReject	New value in CxlRejReason(102), 14 = Quantity Above Account Asset Balance Limit
	Market Data Messages	Added new section regarding Security List Messages workflow
2020-08-03	ExecutionReport	Correction to available balance tags
2020-08-11	SecurityList Request	Add optional tag SecurityGroup(1151)
2020-09-24	V3.10	
	SecurityStatus MarketDataIncrementalRefresh	Added new tag MarketDataID (6006)
2020-10-09	MarketDataIncrementalRefresh	Moved MarketDataID (6006) outside of repeating group
2020-10-21	MarketDataRequestReject	Added new possible value for MDReqRejReason (281) 0=UNKNOWN_SYMBOL
2020-11-16	V3.11	
	SecurityStatus	Added new tag HaltReason (327)
	Supported Time in Force	Day orders expire at the end of the trading session
2021-03-16	V3.12	
	ExecutionReport	Name change for AvailableBalanceData component and its members to PurchasingPowerData
2021-05-06	Weekly Sequence Number Reset	Detail workflow when messages are sent between the initiation of the sequence reset and the client's confirmation
	Message Limits	Improve explanation of message limits
2021-07-12	Order Mass Status Request	Corrected example
	Execution Report	New values in OrdRejReason (103): 17, 21 and 22
2021-08-24	v3.13	
	Execution Report	Added clarification of meaning of OrdRejReason (103) =101
	OrderCancelReject	Added new values for tag Text(58)
	System Status and Security Status	Added Paused state
	NewOrderSingle	Added support for Market Orders

	ExecutionReport: Sent for Order Related Requests	Added support for market orders
	OrderCancelReject	Added new CxlRejReason(102) = 17 (Invalid Order Type)
	Supported Order Types	Added Market orders
2021-10-15	Standard Header	Included TargetSubID (57) and TargetLocationID (143)
2021-10-29	Order Expiry Types	Clarification that Day orders expire at the end of the ErisX Trading Session
2021-11-17	Ticker MarketDataIncrementalRefresh	Enhanced definition of TickerType(7562)
2021-11-19	MarketDataRequest Full Book Aggregated Incremental	Aggregated Incremental subscription is Deprecated
2021-11-29	Message Limits	Addition of messages that are counted towards message limits

2 Introduction

2.1 Purpose

The purpose of this document is to present in detail the Financial Information Exchange (FIX) 4.4 protocol subset available to users of ErisX exchange systems.

3 Product Offering

3.1 Supported Order Types

ErisX supports the following order types:

- **Limit** - An order to buy or sell at a specific price or better.

- **Stop-Limit** - An order that combines the features of a stop order and a limit order. The stop price acts as a trigger to enter a limit order into the market.

- **Market** - An order to buy or sell a stated amount at the current best price.

- **Post-Only** - An order type that provides a user with a way to enter a passive order and guarantee it won't cross the bid-ask spread. This order type has become popular in markets where there is a benefit to being a maker rather than a taker of liquidity.

3.2 Supported Time in Force

The supported time in force values is described in the following table:

Expiry Condition	Description
Day	Orders submitted with this expiry condition that have not been executed will be expired by the system at the end of the ErisX trading session in which they were entered.
Good Till Cancel (GTC)	Orders with this expiry condition remain open and active until either executed or explicitly canceled by the client.
Good Till Date (GTD)	With this time in force, the submitting client specifies the date at which an order is to be expired if not already executed.
Fill or Kill (FOK)	Unless the full quantity of the order can be executed immediately at the specified price or better, an order with this expiry condition will be canceled.
Immediate or Cancel (IOC)	Orders with the expiry condition will be canceled unless a specified minimum quantity can be executed immediately at the specified price or better. Any remaining unfilled quantity is canceled.

3.3 Order Modification

Order parameters such as quantity and expiry condition can be amended on an outstanding order without having to cancel and resubmit the order.

By default, orders that have been partially filled cannot be modified unless the user makes use of the overfill protection logic. See section <u>Overfill protection (New)</u>. A reject message will be received if attempting to modify a partially filled order without the use of overfill protection.

3.4 Minimum Permitted Order Entry Size

There is a minimum permitted order entry size maintained on the ErisX platform. Orders sent for amounts less than the permitted minimum order entry size will be rejected.

3.5 Price Banding

Additional market protection ensures that orders received by the exchange do not surpass a defined threshold and protect members from unexpected fills. Orders outside of the price band will be rejected to prevent an unwanted price movement due to a "fat fingering" of a price. Price bands are set at the Instrument level.

- Order checks are directional; buy orders above the band or sell orders below the band will be rejected.
- Price bands will be configured a number of ticks from a Reference Price.
- Reference Prices follows the following hierarchy:
  - Mid-Price
  - Last Traded Price
  - Best Bid or Best Offer
  - If no trade/bid/offer then Settlement Price
  - If no Settlement Price then Initial Price

3.6 Self Match Prevention

Our Self Match Prevention logic prevents market participants from matching orders within an account, group of accounts with common ownership or FIX Session.

- S.M.P. is enabled by default when an Account is created.
- S.M.P. can be configured on/off by the ErisX Market Operations team.
- S.M.P. can be enabled for an individual account, sub-account or group of FIX credentials under an account.
- If S.M.P is triggered, the resting order will be canceled if S.M.P. is enabled for that account.

4 Futures Specific Functionality

4.1 Regulatory Tags

ErisX requires members to populate some specific FIX tags when sending futures orders to the exchange. Please review the message detail section for the expected values.

The following tags should be sent in the Header of each message:

Tag	Field Name	New Spec
50	SenderSubID	Value used to identify the user that entered the order
142	SenderLocationID	Used to identify the geographical location of the user that entered the order: [Country],[State if in US] eg; US, IL or UK

The following tags MUST be included when entering or modifying an order:

Tag	Field Name	New Spec
581	AccountType	Used to indicate whether an order is for a Customer(1) or House(2) account.
582	CustOrderCapicity	Used to indicate whether the user entering the order is placing it for themselves or for another member. 1 = Member Trading for own account 2 = Clearing firm trading for its Prop Account 3 = Member trading for another member 4 = All other

4.2 Trades which are cleared through a Futures Commission Merchant (FCM)

For users whose trades are cleared through an FCM, an identifier is required to be sent so that those trades are correctly processed by the FCM back office.

Tag	Field Name	New Spec
Component block <Parties>		
448	PartyID	This must contain the FCM Back office account number for the customer of the order.
452	PartyRole	Must contain 24 to indicate that the value in 448 is a Customer Account Reference.

4.3 Instrument details

The <Instrument> component of the SecurityList, SecurityStatus and SecurityDefinition messages include attributes specific to futures contracts:

Tag	Field Name	New Spec
461	CFICode	FCXXSX for futures
167	SecurityType	FUT = Future, SPOT = Spot

200	MaturityMonthYear	Specifies the month and year of maturity (expiry) for a given contract YYYYMM
541	MaturityDate	Specifies the date of maturity (a full date) YYYYMMDD
231	ContractMultiplier	Specifies the ratio of underlying units per future
207	SecurityExchange	Market used to help identify a security = ERSX
Component <EvntGrp>		
864	EvntGrp	The EvntGrp will contain values to indicate the 1st trade date and the last eligible trade data of a contract
865	EventType	5 = Activation (First Trade Date) 7 = Last Eligible Trade Date
866	EventDate	The Date corresponding to the EventType (865) YYYYMMDD

5 Supported Messages

The ErisX FIX specification supports FIX version 4.4 only.

The following convention is used in this document to indicate message direction:

> In: a message type received by ErisX.

> Out: a message type originating from ErisX.

> In/Out: a message type that can be sent to or from ErisX.

Available fields, requirements, values, and their associated meanings are documented in the Message Details section.

Clients are advised to ensure their FIX engine observes the standard FIX 4.4 protocol in which only the order of the first three (3) fields of the header needs to be guaranteed.

5.1 FIX Session Sequence Number Management

FIX Sessions can be maintained across sequential network connections. After an initial session is created, new sessions can continue from the end of the last session by using the last outbound sequence number. On reconnect, clients can use the logon confirmation message sequence number to detect a gap since the last received message. If the client detects a gap, the client application can request all missed messages using a ResendRequest.

All available requested messages will be resent with updated SendingTime(52), OrigSendTime(122), PossDupFlag(43) field set to 'Y' and recalculated CheckSum value.

A gapFill message will be sent in lieu of the administrative messages or when messages are no longer available. Clients should avoid submitting subsequent ResendRequest messages. This will simply replace the prior ResendRequest resulting in a delay of normal processing.

Note: If a client application receives a resend request FROM ErisX Match Engine, the client application should respond with a SequenceRest(35=4) message and include GapFillFlag

(123=Y). Client applications should never resend any business messages that have been previously sent.

5.2 Weekly Sequence Number Reset

With the goal of enabling 24/7 FIX sessions, ErisX has implemented a weekly Sequence Number Reset workflow. At the same moment, password expiration checks will also take place.

At **14:00 CT every Sunday**, the following message flow takes place:
- ErisX will send a Logon(**A**) message with ResetSeqNumFlag(**141**)=**Y** and MsgSeqNum(**34**) = **1**. The other tags and values will correspond to the last known Logon(A) message previous to the reset. This message will be sent to all Logged On FIX sessions on all three kinds of FIX gateways (Order Management, Market Data and Drop Copy).
- The user should process the message by resetting the sequence numbers and responding with a Logon(**A**) message with ResetSeqNumFlag(**141**)=**Y** and MsgSeqNum(**34**) = **1**.
- Upon receiving the Logon(A) message from the client, ErisX will send a TradingSessionStatus(**h**) message to the client with the new sequence number: MsgSeqNum(**34**) = **2***.

* In case new messages from the Exchange arrive at the FIX Gateways between the time when ErisX sends the Logon(**A**) message with the ResetSeqNumFlag(**141**)=**Y** and MsgSeqNum(**34**)=**1** and the time the client session responds with the appropriate Logon(**A**) message, ErisX will send those messages with the new sequence numbers (starting at MsgSeqNum=2 and incrementing from there). After receiving the Logon(**A**) message from the client, ErisX will send a TradingSessionStatus(**h**) message to the client with MsgSeqNum(**34**) equal to the next available sequence number from the last message sent to the client (MsgSeqNum >= 2).
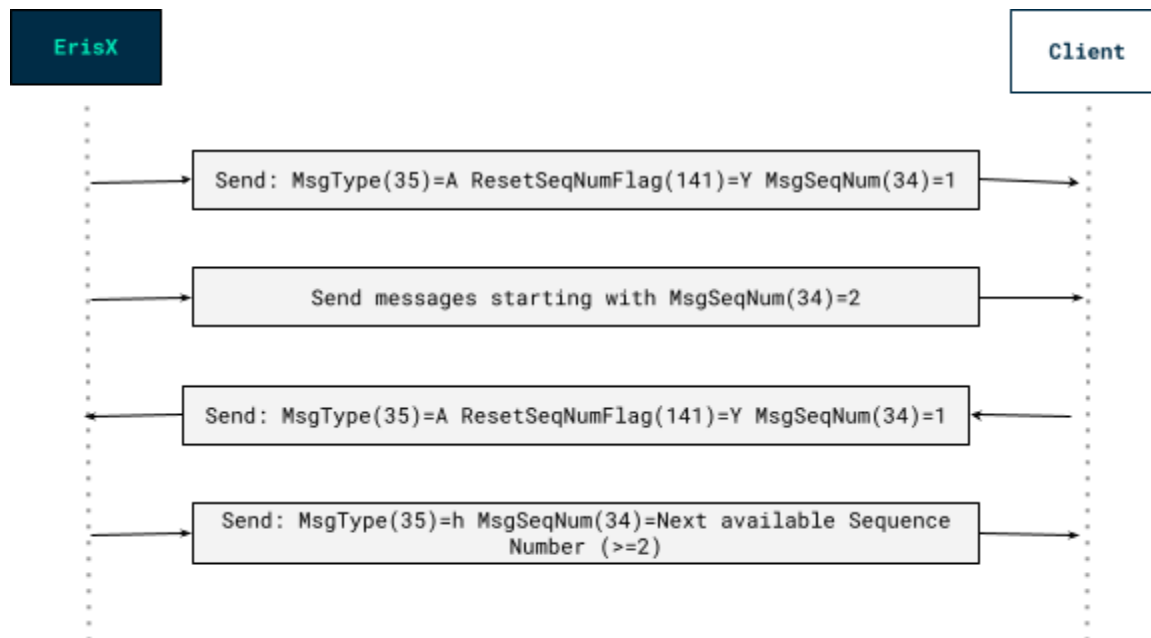
At **14:00 CT on every Sunday**, ErisX will perform a check for expired passwords. This check will apply to FIX IDs from all three gateways (Order Management, Market Data and Drop Copy). Any logged on FIX session whose **password is expired will be disconnected**. Upon next Logon, the user must reset the password before any application messages can be sent/received. For additional information on resetting the password, please see [User Request (change password)](#)..

5.3 Session Messages

Session messages establish, maintain and terminate an ErisX connection.

- Logon – (In/Out) message sent to initiate a FIX session to ErisX. The Logon message establishes the communication session, authenticates the connecting client, and initializes the message sequence number.

- Heartbeat – (In/Out) message sent by ErisX during periods of application inactivity to ensure connection validity. The receiving party should always respond with a heartbeat message.

- Resend Request – (In/Out) request that certain messages be resent. Often used when gaps are detected in the sequence numbering, when a message is lost, or during the initialization process.

- Test Request – (In/Out) used to verify session connectivity and to synchronize sequence numbers. The receiving party should always respond with a heartbeat message.

- Logout – (In) signals the normal termination of the trading session. A session terminated without a Logout message will be considered an abnormal condition. The ErisX FIX gateway treats a session as logged out whenever the communication session is dropped.

5.4 Timestamping / TransactTime (tag 60)

Messages sent by client applications will need to include TransactTime (60). The system will validate the value sent down to one second precision and accuracy.

Responses from the match engine will include TransactTime (60) and will be sent with nanosecond precision. YYYYMMDD-HH:MM:ss.SSSSSSSSS.

The timestamp on outgoing messages will represent the time the corresponding message was received by the FIX gateway that resulted in the update.

5.5 Application Messages

Once a proper session is established, application messages are used to receive market data, to submit orders, and to receive execution reports.

Messages:

- **Trading Session Status** – (Out) application message sent from ErisX indicating the trading session is fully initialized. New application messages should not be sent until receipt of this message with a TradSesStatus of System Ready. A Business Message Reject will be received for any application messages sent prior to receiving this message.

- **Business Message Reject** – (Out) application message sent in response to any application-level message that cannot be replied to with a normal matching response message. For example, ErisX sends it when an application-level message is received prior to a Trading Session Status message having been sent.
Also sent when a request message is received during non-ErisX trading hours. For a schedule of non-trading hours, please contact your ErisX member service representative.

- **Market Data Request** – (In) message is used to subscribe/unsubscribe to market data. Each request message must contain one requested instrument type. Repeating instrument requests are not supported at this time.

- **Market Data Request Reject** – (Out) message is sent to indicate a Market Data Request message cannot be processed; e.g., due to the system being down, no permission, or system off-hours.

- **Market Data (Incremental Refresh)** – (Out) message sent in response to a Market Data Request message. This message contains entries for one pair only. It can contain both bid and offer updates or show an aggregated content of the book where the total number of orders is shown.

- **Security List Request** – (In) message sent to request Instrument Info for all Instruments configured.

- **Security List** – (Out) message sent in response to a Security List Request message. It contains Instrument information for all configured Instruments.

- **New Order (Single)** – (In) message sent to input an order into the ErisX trading system.

- **Execution Report** – (Out) message returned in response to a New Order, the completion of an order, the partial fill of an order, an order cancel request, an order replace request or an order status request. In each case, the Execution Report will show the current state of the order in question.

- **Order Cancel/Replace Request** – (In) message sent to amend an outstanding order. An Order Cancel Reject message will be sent if the requested order cannot be replaced. An Execution Report with the appropriate execution type will be immediately sent for all other conditions.

- **Order Cancel Request** – (In) message sent to cancel a particular order. If an order has been partially filled, only the outstanding amount can be cancelled. Also used to cancel all outstanding orders.

- **Order Cancel Reject** – (Out) message sent when the request to cancel or cancel/replace an order cannot be processed.

- **Order Mass Order Status Request** - (In) message used to request a current list of the working orders.

15

6 Message Workflow

6.1 General Workflow

A successful logon is required before further messages are sent and the session must receive Trading Session Status messages with TradSesStatus (340) = 101, System Ready. The Trading Session Status message will normally be received immediately after logon, if the previous session terminated abnormally, a few seconds delay can be encountered while cleanup operations are performed.

ErisX will respond to any application-level messages received prior to it having sent a Trading Session Status message with a Business Message Reject.

6.2 System Status and Security Status

We have refactored the behavior of TradingSessionStatus and added a new message called SecurityStatus.

TradingSessionStatus refers to technical system availability. The only supported values are:

- 101 - System Ready, which indicates that the system is available for technical use, but not necessarily reflects if a market is tradable or not.
- 105 - System Disconnect, which indicates that the system will go down for maintenance.

All other values have been deprecated in favor of the Security Status message described below.

The new message, SecurityStatus is a business level message which indicates the tradable state of an instrument. This is the message your systems should look at in order to determine if orders could be placed into the market and matched.

The new security status values are:

- Open: This indicates that continuous trading is available
  - All order management messages are allowed (New, Cancel, Modify)
- Closed: This indicates that the order book is not available for trading. No order management is available. All New, Cancel and Modify will be rejected
- Paused: This indicates that trading has been paused. Only Cancel messages are permitted

- Pre-Open (When Futures Launch): This indicates that orders can be placed, but no matching will occur due to the instrument being in an auction state. This security state is to allow for price discovery and orderly re-opening of markets.

Security status messages are sent in real-time as instruments transition between security status states. This message is also sent upon subscribing to an instrument's market data.

6.3 Passwords for Logon Messages

The ErisX FIX service requires users connecting via the FIX protocol to present a password as part of the Logon (35=A) message. To successfully connect, a user needs to set the password

field, tag 554, a ErisX customized tag, to the valid password associated with the id specified in tag 49.

A Logon message containing an invalid password will be rejected. The ErisX response to a valid Logon message will not contain tag 554.

It is recommended that users change their session passwords after receiving them from the ErisX Client Services team.

An expired password must be reset before any application messages, e.g., market data requests, new orders entry, etc. will be processed. Upon successful Logon, if a password needs to be reset, a TradingSessionStatus (35=h) message will be returned with tag 340=1 and tag 58 set to 'Password reset is required.' Until the password is reset, all further application messages will be rejected by the ErisX FIX gateway.

An account is locked after five (5) invalid password logon attempts. A locked account must be unlocked by the ErisX support desk before a user can logon successfully again.

As long as the account is not locked, its password can be reset by the user at any time by sending the appropriate UserRequest (35=BE) message.

7 Market Data Messages

Market data messages are published in three functional groups:

1. **Trades**
   - Trade information is batched into one message grouped by price for a given aggressor if possible.
   - Trades for the same match event may be split across multiple incremental updates due to message size limits.
   - The final message for the match event will contain a Custom tag EventIndicator (6001=1) representing the end of trades.

2. **Statistics**
   - Statistical information will be published for an instrument to inform users of such changes to things like Session High, Session Low, and Total Volume.

3. **Book Updates**
   - Book update information will be batched into one message for a given aggressor if possible.
   - Updates for the same match event may need to be split across multiple incremental messages .
   - Custom tag EventIndicator (6001=2) will be sent on the final message of a sequence to indicate that all prior messages were part of an atomic matching event. The value of 2 is referred to as EndOfEvent.

Example:

If we consider the following orderbook (broken down to individual orders to show granularity).

BID Q	BID	ASK	ASK Q
10	9002	9010	50
10	9002		
5	9002		
5	9001		
5	9001		
15	9000		

When: An order is placed to sell 50 BTC/USD @ 9000

The market data messages would be as follows:

Trades Incremental (35=X)

```
268=3 (NoMDEntries=3)
      269=2 (Trade) |55=BTC/USD|270=9002|271=25|346=3 (NumberOfOrders=3)
      269=2 (Trade) |55=BTC/USD|270=9001|271=10|346=2 (NumberOfOrders=2)
      269=2 (Trade) |55=BTC/USD|270=9000|271=15|346=1 (NumberOfOrders=1)
6001=1 (EventIndicator=EndOfTrade)
```

Statistics Incremental (35=X)

```
269=8 (SessionLow) |55=BTC/USD|270=9000
```

Statistics Incremental (35=X)

```
269=B (TotalVolume) |55=BTC/USD|271=50
```

BookUpdates Incremental (35=X)

```
268=6 (NoMDEntries=6)
      269=0 (Bid) 279=2 (Delete) |55=BTC/USD|270=9002|271=10
      269=0 (Bid) 279=2 (Delete) |55=BTC/USD|270=9002|271=10
      269=0 (Bid) 279=2 (Delete) |55=BTC/USD|270=9002|271=5
      269=0 (Bid) 279=2 (Delete) |55=BTC/USD|270=9001|271=5
      269=0 (Bid) 279=2 (Delete) |55=BTC/USD|270=9001|271=5
      269=0 (Bid) 279=2 (Delete) |55=BTC/USD|270=9000|271=15
6001=2 (EventIndicator=EndOfEvent)
```

7.1 Market Data Subscriptions

If the SubscriptionRequestType equals 1, snapshot plus updates, ErisX continuously sends new updates to the client and reports when a price is no longer available. Three fields affect the subsequent price updates:

- MarketDepth
- MDUpdateType
- AggregatedBook

For MDUpdateType, ErisX supports the following incremental update request types:

- FullBook – aggregate incremental
- FullBook – non-aggregate incremental

The workflow for each possible request type is described in the following sections.

A Market Data Request message can be rejected. See MarketDataRequestReject message for possible values.

7.2 FullBook Aggregated Incremental (**DEPRECATED**, Use Non-Aggregated instead)

- A complete aggregated book is sent to the client. If more than one order exists at the same price in the same instrument, only one price will be displayed with the quantity amounts aggregated into one.

- An update from the server either cancels an outstanding price with the same MDEntryID (278) or effectively cancels and replaces it with a new price.

7.3 FullBook Non-aggregated Incremental

- A complete non-aggregated book is sent to the client.

- An update from the server either cancels an outstanding price with the same MDEntryID (278) or effectively cancels and replaces it with a new price.

7.4 Handling MDEntryID

All market data messages are associated with an MDEntryID (278) tag that identifies the price to remove or replace in a full book scenario.

The MDEntryID (278) is unique per instrument within a single session represented as a hexadecimal encoding of a long data type as a string.

Within the same symbol, only one (1) price can be outstanding for any MDEntryID (278), and subsequent updates having the same MDEntryID (278) as an outstanding price replace it or delete it from the book. The action is specified in MDUpdateAction (279): 0 = new and 2 = delete.

The client session is responsible for monitoring the MDEntryID (278) tag to keep track of these updates.

7.5 Re-subscriptions

Market data subscriptions are session-based and are not permanent. A session must re-subscribe to the instruments it is interested in receiving on each new connection.

7.6 Trade Ticker

The ErisX FIX market data service supports a trade ticker which reports executed trade updates, a.k.a. a ticker containing the instrument, executed price, quantity, and the ticker type Aggressive or Passive.

Paid or given is determined from the perspective of the aggressed order. An order that aggresses or 'hits' a bid price will appear in the ticker feed as given. An order that aggresses or 'lifts' the offer price will appear in the ticker feed as paid.

7.7 Security List Messages

SecurityList message as a response to a SecurityListRequest, may be broken down into several consecutive messages as a result of a large number of available instruments. In that case, the user should expect to receive multiple SecurityList messages, each with different instruments. The last SecurityList message in the sequence will have tag populated with **LastFragment(893)=Y** to indicate that no more SecurityList messages are expected.

8 Order Processing Messages

Refer to the Message Details section for a complete listing of message types and associated tags.

8.1 Order Input and Execution

A New Order Single message is used to place orders into the ErisX system. ErisX will reply with an Execution Report message, which indicates whether or not the order has been accepted. Execution reports are also sent when there is a change in an order's status, e.g., when an order is filled, modified or canceled.

Message	From	To	Message Detail
New Order	Client	ERSX	35=D
Execution Rpt (Ack)	ERSX	Client	35=8 with 150=0 and 39=0
Order Cancel Request	Client	ERSX	35=F
	ERSX	Client	35=8 with 150=4 and 39=4 if successful 35=9 with 39=8 if rejected
Order Replace (Modify)	Client	ERSX	35=G
	ERSX	Client	35=8 with 150=5 and 39=5 if successful 35=9 with 150=8 and 39=8 if rejected
Order Fill			*Unsolicited*
Execution Rpt (Fill)	ERSX	Client	35=8 with 150=F and 39=1 (Partial fill) or 39=2 (Filled)
Order Mass Status Request	Client	ERSX	35=AF
Execution Rpt	ERSX	Client	35=8 will be sent for all currently working orders

8.2 Order Expiry Types (Time in Force)

By default all orders are "Day" orders, meaning if they have not already expired or canceled, they will automatically expire at the end of the ErisX trading session. Clients can set different expiry conditions through the TimeInForce (59) field.

Note**:** Previous versions of the match engine responded to new orders with TimeInForce = 0 (day order) with a TimeInForce = 6 (Good till date). This will no longer be the case and day orders will be responded to with TimeInForce = 0 (Day order)

- The ErisX trading day ends at 16:00:00 (CST/CDT).
- Any day order received at 16:00:00 (CST/CDT) will be in effect until 16:00:00 the next business day.

A TimeInForce = 6 must also have the ExpireDate (432)

8.3 Execution Reports

ErisX sends Execution Report messages to:

- Confirm the receipt of an order
- Confirm changes to an existing order
- Reply to order status messages
- Relay order fill information on active orders
- Relay order status change information
- Reject an order

In a normal workflow, after sending an Execution Report message to indicate the receipt of the order, ErisX may continue to send one or more Execution Report messages to relay order fill information if applicable. If the order is filled in full, it will be indicated in the Execution Report. In cases of partial fills, ErisX will send Execution Report messages indicating partial fills until the order is completely filled, the client actively cancels the remaining portion of the order, or the remaining portion expires.

Two fields in the Execution Report message warrant explanations, the ExecType (150) and the OrdStatus (39). For a multiple filled order, the ExecType (150) field reports information on the individual fill and the OrdStatus (39) field reports information on the overall order status.

An Order Replace Request message is used to update an active order. An Order Cancel Request message is used to cancel an order or any remaining portion of an order. ErisX immediately responds to both with an Execution Report confirming or rejecting the request.

8.4 Handling Fill Type Execution Reports

Both the ExecType (150) and OrdStatus (39) fields need to be examined to properly process an Execution Report.

The ExecType (150) indicates the status of the current action on an order.

The OrdStatus (39) indicates the overall status of the order.

ExecType (150) = F, and OrderStatus (39) = 1 indicates a partial fill.

When ExecType (150) = F and OrderStatus (39) = 2, the order has been completely filled.

8.5 On Behalf of Routing

Trades done on ErisX are booked under the account that entered the trade and the ErisX legal entity under which the account exists.

A client with multiple trading accounts can be configured to trade on behalf of each account via a single FIX connection. The appropriate account is indicated on the NewOrderSingle message by setting PartyID (448) to a FIX reference that is tied to the target account and setting PartyRole (452) to '3'. The trade, if done, will then be booked under the corresponding account.

The Account field on cancel, replace, and order status request messages must match those used in the original NewOrder message, if any.

On behalf of mappings must be defined on ErisX prior to use via ErisX client services.

8.6 Order Update and Replace

It is possible to update or replace an outstanding order without first having to cancel it.

The following NewOrder message fields can be updated:

- OrderQty (38) – Specified amount
- MinQty (110) – Cannot be greater than the value of OrderQty (38)
- Price (44) – Limit price
- StopPX (99) – can be modified for OrdType (40) = 3.
- TimeInForce (59) – Expiration type: if tag 59= 6, then conditionally required field ExpireDate (432) can be changed. The tag 59 value itself cannot be changed.

Note: Order replace requests on filled or partially filled orders will be rejected unless the overfill protection functionality tag 5000 is used.

8.7 Overfill protection (New)

If an order has been partially filled, then our custom tag OverfillProtection (5000=Y or N) must be included on the 35=G Order Replace message.

- With Overfill Protection = Y, the original quantity is modified which will update the remaining quantity (LeavesQty) to the new requested qty minus the already filled cumulative quantity.
- Whereas with Overfill Protection = N, the remaining quantity (LeavesQty) is set to the new quantity as specified in the modified message.
- If the Overfill Protection tag 5000 is not set and the order which is requesting modification has been partially filled, then the request will be rejected.

Example:

Given: An original order to buy 5 lots which has been partially filled.

```
Order Quantity = 5, Filled = 3, LeavesQty = 2, Cancelled = 0
```

When: A modify request is received containing an OrderQty of 4 with Overfill Protection = **Y**

Then: The order quantity is set to 4, which reduces the remaining quantity (LeavesQty) quantity down to 1.

```
Order Quantity = 4, Filled = 3, LeavesQty = 1, Cancelled = 0
```

Or:

When: A modify request is received containing an OrderQty of 4 with Overfill Protection = N

Then: The remaining quantity (LeavesQty) is set to 4

```
Order Quantity = 7, Filled = 3, LeavesQty = 4, Cancelled = 0
```

8.8 Order Cancel or Replace Using the Client Assigned Order ID

By default, to cancel an outstanding order, regardless of type, a client must specify three (3) tags:

- OrderID (37) – ErisX assigned order id
- OrigClOrdID (41) - Client assigned order id
- ClOrdID (11) – Client assigned id for the replacement order or the cancel request

8.9 Cancel of Complex Orders

The OrdType (40) tag must be explicitly specified in any complex order cancel request, where complex orders are defined as:

- 4 - Stop-Limit order

If OrderType (40) is not specified, an execution report will be returned containing tag "58=UNKNOWN ORDER - [order id]"

8.10 Cancel all Orders

Client applications are able to cancel all working orders for a FIX session via a single OrderCancelRequest (35=F) by sending "OPEN_ORDER" in tags ClOrdID (11) and OrigClOrdID (41) as well as including OpenOrders = Y (7559).

FIX Sessions that are used to route orders 'on behalf of' other FIX sessions will need to include the Parties Component block and the valid PartyID (448) of the FIX sessions which the orders were routed for.

8.11 Mass Order Status Request

The mass order status request should be used to obtain a set of working orders for a given session. Tag 912=Y will indicate when the last message in the response has been received.

Members who are disconnected should use the resend request message logic to restate working orders.

If no working orders are found, 911 = 0 and an empty Execution Report will be sent with order-related fields being 0 or "NA".

A FIX Session may use the Mass Order Status request to take ownership of the working orders of another authorized FIX Session configured as a routee. For this purpose, the FIX session that is taking ownership will send a Mass Order Status Request, specifying the partyID of the FIX session whose orders it wants to take ownership of. After processing the corresponding Mass Order Status Request, only the new FIX session will be allowed to interact with the working orders.

Example:

PartyID = A is configured as a routee of PartyID = B. PartyID=A sends a new limit order and the order rests in the order book with id=1. PartyID=B sends a Mass Order Status Request specifying PartyID=A, thus, taking ownership of PartyID=A working order with id=1. Now PartyID=B can interact with the order with id=1, but PartyID=A is no longer able to interact with that order as ownership has already changed.

Note: Previous versions of this request were responded to with a full set of order history. This has been removed in our new match engine.

8.12 ErisX Session Logouts and Disconnects

FIX trade ids are configured so that an id's outstanding orders will be canceled upon a logout or an unplanned session termination. This feature can be turned off upon request so that outstanding orders will not cancel when a session ends.

8.13 Message Limits

The purpose of messaging throttle limits is to prevent excessive messaging on the exchange that could have negative effects on all users.

- Each FIX ID will have a throttle limit per time interval:
  - Max messages per X seconds
    - Order Entry messages:

- New Order Single, Cancel Replace Order, Order Mass Status Request, Order Cancel Request*.
  - Market Data messages:
    - Security List Request.
  - Drop Copy messages:
    - Trade Capture Report Request.
  - Administrative messages:
    - Logon, Logoff, User Request, Resend Request, Sequence Reset, Test Request, Heartbeat.
- A FIX ID will be disconnected when it breaks a throttle limit.
- The interval time starts when the first message is received.
- At the end of the time interval, the message counter is reset.
  - This is not a rolling interval

* Members are encouraged to use Cancel All when attempting to cancel all their open orders instead of sending individual cancel requests for each order to avoid hitting message limits.

If a session breaches limits, ErisX will send a TradingSessionStatus message with TradSesStatus (340) = 105, indicating the session will be closed. Orders in process will be permitted to complete, no new order or order cancel replace messages will be accepted. Once all order processing is completed, ErisX may log the session out with tag 58="message limit exceeded".

To avoid unexpected fills, upon receipt of a Trading Session Status message with tag 340=105, ErisX advises users to immediately cancel all outstanding orders and then to log off as no new order or cancel replace operations will be permitted.

If the user id is enabled for "Cancel Orders on Disconnect," their orders will be canceled by ErisX at the time of the Logout. However, during the logout process orders can be matched and filled.

9 Message Details

In the tables below, specific messages are presented in columns: "Tag", "Field Name," "Required", "New Spec" and "What's Changed?".

As we update our FIX specification the "New Spec" field will contain the updated information relevant to the change. Users should review the "What's Changed" column to quickly review changes between versions and be proactive in testing changes to make sure they do not affect operations.

Order related messages follow the same table format as used except that the "Required" column values are relative to the value specified in OrdType (40) field.

Note the following conventions:

The values under the "Required" column indicate one of the following:

- 'Y' – field is mandatory and must be sent or received as a part of the message.
- 'N' – Non-required field that should be omitted unless directed otherwise by ErisX.
- 'NA' – field is not used at all in the context for the message.

9.1 Session Messages

9.1.1 Standard Header

Tag	Field Name	Rqd	New Spec	Updated?
8	BeginString	Y	Message start. Handled by FIX engine.	
9	BodyLength	Y	Message length. Handled by FIX engine.	
35	MsgType	Y	The message type. Refer to individual messages for valid values.	
49	SenderCompID	Y	Provided by ErisX - the user's trading account id.	
56	TargetCompID	Y	Default setting is "ERISX"	
34	MsgSeqNum	Y	Message sequence number. Handled by FIX engine.	
122	OrigSendingTime	N	Original time of message transmission (always expressed in UTC (Universal Time Coordinated) when transmitting orders as the result of a resend request.	
50	SenderSubID	F	Value used to identify the user that entered the order	
142	SenderLocationID	F	Used to identify the geographical location of the user that entered the order [Country],[[State if in US] eg; US,IL or UK	
57	TargetSubID	F	On messages sent to the client, value used to identify the user that entered the order. I.e. what was entered in SenderSubID (50) tag in the request message	Added v3.13
143	TargetLocationID	F	On messages sent to the client, used to identify the	Added

Tag	Field Name	Rqd		Updated?
			geographical location of the user that entered the order: [Country],[State if in US] eg; US, IL or UK. I.e. what was entered in the SenderLocationID (142) tag in the request message	v3.13
43	PossDupFlag	N	Indicates possible retransmission of message with this sequence number: Y = Possible duplicate N = Original transmission Used on SequenceResets and ResendRequests.	
52	SendingTime	Y	The GMT timestamp on the message.	

9.1.2 Standard Trailer

Tag	Field Name	Rqd	New Spec	Updated?
10	CheckSum	Y	A value calculated by the FIX engine from the message data and transferred with the data. If the data received does not match the CheckSum value, the data was corrupted in transit.	

9.1.3 Logon

Tag	Field Name	Rqd	Comments	Updated?
	Standard Header	Y	**MsgType tag 35=A**	
98	EncryptMethod	Y	0 – not encrypted is the only accepted value.	
108	HeartBtInt	Y	Heartbeat interval in seconds.	
141	ResetSeqNumFlag	N	Y – Resets both incoming and outgoing sequence numbers to 1	
554	Password	Y	Password for the given FIX ID	
	Standard Trailer	Y		

9.1.4 Logout

Tag	Field Name	Rqd	Comments	Updated?
	Standard Header	Y	**MsgType tag 35=5**	
58	Text	Y	Possible values for initial logon failure include: Configuration Error, System Failure, Authentication Error	
	Standard Trailer	Y		

9.1.5 Resend Request

Tag	Field Name	Rqd	New Spec	Updated?
	Standard Header	Y	**MsgType tag 35=2**	
7	BeginSeqNo	Y	First sequence number in the range to be resent.	
16	EndSeqNo	Y	Last sequence number in the range to be resent. For single message resend requests, set BeginSeqNo = EndSeqNo.	

28

			If request is for all messages subsequent to a particular message, EndSeqNo = 0.	
Standard Trailer	Y			

9.1.6 Test Request

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType tag 35=1**	
112	TestReqID	Y	Unique ID of test request.	
Standard Trailer		Y		

9.1.7 Heartbeat

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType tag 35=0**	
112	TestReqID	N	Required if a heartbeat is due to a Test Request message.	
Standard Trailer		Y		

9.1.8 User Request (Change Password)

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType 35=BE**	
923	UserRequestID	Y	Unique identifier for User request	
924	UserRequestType	Y	3 = ChangePasswordForUser, only valid value	
553	Username	Y	The SenderComp ID of the password to be changed	
554	Password	Y	Current Password or passphrase	
925	NewPassword	Y	New Password or passphrase	
Standard Trailer		Y		

9.1.9 User Response

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType 35=BF**	
923	UserRequestID	Y	Unique identifier for User request	
553	Username	Y	The SenderComp ID of the password to be changed	
926	UserStatus	Y	Indicates the status of the user. Valid values are 3 = User not recognized 5 = UserPasswordChanged 6 = Other	
927	UserStatusText	N	A text description associated with a user status	
Standard Trailer		Y		

9.2 Application Messages

9.2.1 TradingSessionStatus

Tag	Field Name	Rqd	New Spec	Updated?
	Standard Header	Y	**MsgType tag 35=h**	
336	TradingSessionID	Y	Identifier for this trading session.	
340	TradSesStatus	Y	Only valid values: 101=System Ready 105=System Disconnect	
58	Text	N	Descriptive text message.	
	Standard Trailer	Y		

9.2.2 BusinessMessageReject

Tag	Field Name	Rqd	New Spec	Updated?
	Standard Header	Y	**MsgType tag 35=j**	
45	RefSeqNum	N	MsgSeqNum of rejected message.	
372	RefMsgType	Y	The MsgType of the FIX message being rejected.	
371	RefTagID	Y	The tag number of the FIX field being referenced. Only sent when 'Business Message Reject' message is generated by the FIX engine.	
373	SessionRejectReason	Y	Code to identify reason for a session-level reject message. Only sent when 'Business Message Reject' message is generated by the FIX engine.	
380	BusinessRejectReason	Y	Code to identify reason for this reject message. 0 = Other 1 = Unknown ID 2 = Unknown Security 3 = Unsupported Message Type 4 = Application not available 5 = Conditional Required Field Missing 6 = Invalid Logon	
	Standard Trailer	Y		

9.2.3 SecurityListRequest

Tag	Field Name	Rqd	New Spec	Updated?
	Standard Header	Y	**MsgType tag 35=x (lowercase)**	
320	SecurityReqID	Y	Unique security request ID.	
559	SecurityListRequestType	Y	0 = Symbol	

	1151	SecurityGroup	N	ALL: ErisX will return all active instruments Other value, ErisX will return all active instruments where securityGroup matches the requested value If securityGroup is not specified, ErisX will only return a default subset of contracts	V3.9
Component <Instrument>			Y		
	55	Symbol	Y	Set to NA	
	460	Product	Y	2 = Commodity	
End Component <Instrument			Y		

9.2.4 SecurityList

Tag		Field Name	Rqd	New Spec	Updated?
Standard Header			Y	**MsgType tag 35=y (lowercase)**	
320		SecurityReqID	Y	Unique security request ID	
322		SecurityResponseID	Y		
560		SecurityRequestResult	Y	0 = ValidReq, 1 = InvalidReq	
Component <SecListGrp>			Y		
146		NoRelatedSym	Y		
	Component <Instrument>				
→	55	Symbol	Y	Instrument (E.g. BTC/USD)	
→	460	Product	Y	2 = Commodity	
→	461	CFICode	F	FCXXSX for futures	
→	167	SecurityType	F	SPOT = Spot, FUT = Futures	
→	231	ContractMultiplier	F	The quantity of underlying units per 1 futures contract	
→	200	MaturityMonthYear	F	Specifies the month and year of maturity (YYYYMM)	
→	541	MaturityDate	F	Specifies date of maturity (YYYYMMDD)	
→	207	SecurityExchange	F	Market used to help identify a security = ERSX	
Component <EvntGrp>					
→	864	NoEvents	F	Number of repeating EventType entries	
→	865	EventType	F	5 = First Trade Date 7 = Last Trade Date 8= Last Trade Time 9= Expiry Time	Updated V3.6
→	866	EventDate	F	Date of event YYYYMMDD	Updated V3.6

	868	EventTime	F	Time of event in UTC XX:XX:XXZ	Added V3.6
→					
End Component <EvntGrp>					
→	969	MinPriceIncrement	Y	Minimum price change for a given symbol	
→	107	SecurityDesc	Y	Security description	
→	562	MinTradeVol	N	The minimum order quantity that can be submitted for an order	
→	1140	MaxTradeVol	N	The maximum order quantity that can be submitted for an order	
→	561	RoundLot	N	Trading lot size of security (minimum fill size)	
→	15	Currency	Y	This will be the Base currency	
→	1151	SecurityGroup	N	An exchange specific name assigned to a group of related securities which may be concurrently affected by market events and actions.	Added V 3.9
→	6005	ProductCode	N	Groups asset based on a common contract specification	Added V 3.9
→	1199	Cap	N	Upper Price Boundary of a contract	Added V 3.9
→	1200	Floor	N	Lower Price Boundary of a contract	Added V 3.9
		End Component <Instrument>			
End Component <SecListGrp>			Y		
	893	LastFragment	N	Y = Indicates that no more SecurityList messages are expected	Added V 3.9
Standard Trailer			Y		

9.2.5 SecurityStatus

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType tag 35=f (lowercase)**	
324	SecurityStatusReqID	Y	Unique security request ID (value provided in tag 320).	
Component <Instrument>		Y		
55	Symbol	Y	Instrument (E.g. BTC/USD)	
460	Product	Y	2 = Commodity	
461	CFICode	F	FCXXSX for futures	
167	SecurityType	F	FUT = Future	Updated V3.4
107	SecurityDesc	Y	E.g. Bitcoin USD	
200	MaturityMonthYear	F	Specifies the month and year of maturity(YYYYMM)	
207	SecurityExchange	F	Market used to help identify a security = ERSX	

541	MaturityDate	F	Specifies date of maturity (YYYYMMDD)	
969	MinPriceIncrement	Y	Minimum price increase for a given Instrument	
231	ContractMultiplier	F	Specifies the ratio of underlying units per future	
562	MinTradeVol	N	The minimum order quantity that can be submitted for an order.	
1140	MaxTradeVol	N	The maximum order quantity that can be submitted for an order	
561	RoundLot	N	Trading lot size of security (minimum fill size).	
15	Currency	Y	This is the base Currency.	
6006	MarketDataID	N	Sequence number which uniquely identifies all unsolicited market data messages within a trade date, for example 35=X and 35=f. Messages containing the same Global Market Data ID within a Trade Date should be considered as duplicates.	V3.10
End Component <Instrument>				
326	SecurityTradingStatus	Y	2=Trading Halt 15=New Price Indication(Opening) 17=Ready To Trade (Open) 18=Not available for Trading / End of Session (Close) 21=Pre Open 28=Pre Close	
327	HaltReason	N	Present when 326=2 and Instrument is set to Paused. 327=X EQUIPMENT_CHANGEOVER	V3.11
1174	SessionEnd	N	4 = Change of Trading Session	
Standard Trailer		Y		

9.2.6 SecurityDefinition

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType tag 35=d (lowercase)**	
320	SecurityReqID	Y	Unique security request ID.	
322	SecurityResponseID	Y		
560	SecurityRequestResult	Y	0 = Valid Request 1 = InvalidReq	
Component <Instrument>		Y		
980	SecurityUpdateAction	Y	A = Add D = Delete M = Modify	
1682	MDSecurityTradingStatus	N	2 = Trading Halt 15 = New Price Indication (Opening)	

			17 = Open 18 = Closed 21 = Pre Open 28 = Pre Close	
55	Symbol	Y	Instrument (E.g. BTC/USD)	
460	Product	Y	2 = Commodity	
461	CFICode	Y	FCXXSX for futures	
167	SecurityType	Y	FUT = Future SPOT = Spot	
107	SecurityDesc	Y	E.g. Bitcoin USD, Litecoin USD, ... Litecoin BTC, Bitcoin Cash BTC...	
200	MaturityMonthYear	N	Specifies the month and year of maturity (expiry) for a given contract (YYYYMM)	
207	SecurityExchange	F	Market used to help identify a security = ERSX	
541	MaturityDate	N	Provides the full date of the maturity (expiry) for a given contract (YYYYMMDD)	
969	MinPriceIncrement	N	Minimum price change for a given symbol	
231	ContractMultiplier	N	Indicates the quantity of underlying units per 1 futures contract	
562	MinTradeVol	N	The minimum order quantity that can be submitted for an order	
1140	MaxTradeVol	N	The maximum order quantity that can be submitted for an order	
561	RoundLot	N	Trading lot size of security (minimum fill size).	
Component <EvntGrp>				
864	EvntGrp	F	Event Group used to describe futures trading cycle	
865	EventType	F	5 = Activation (First Trade Date) 7 = Last Eligible Trade Date	
866	EventDate	F	The Date corresponding to the EventType (865) (YYYYMMDD)	
End Component <EvntGrp>				
15	Currency	Y	This will be the Base currency	
End Component <Instrument>		Y		

9.2.7 MarketDataRequest

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType tag 35=V**	

Tag	Field Name	Rqd	New Spec	Updated?
262	MDReqID	Y	A unique ID assigned by the client to the Market Data Request. To unsubscribe from market data, the same ID must be sent with tag 263 = 2.	
263	SubscriptionRequestType	Y	Specifies the data request type. A Snapshot + Updates request is for the current state of the market and all subsequent updates. Valid values: 1 = Snapshot + Updates (Subscribe) 2 = Unsubscribe	
264	MarketDepth	Y	Depth of market for Book Snapshot. 0 = Full Book	
265	MDUpdateType	N	Required if SubscriptionRequestType = Snapshot + Updates (1): 1 = Incremental Refresh	
266	AggregatedBook	Y	N = Non-aggregate Y = Aggregate	
Component <MdReqGrp>		Y		
267	NoMDEntryTypes	Y	Number of MDEntryType fields being requested. 2 = bid and offer	
→ 269	MDEntryType	Y	Market Data entries types list: 0 = Bid 1 = Offer Repeated field: 269=0, 269=1	
End Component <MdReqGrp>		Y		
Component <InstrmtMDReqGrp>		Y		
146	NoRelatedSym	Y	Number of related symbols in the request. This value is always equal to '1' in current version.	
→ 55	Symbol	Y	Instrument (E.g. BTC/USD)	
End Component <InstrmtMDReqGrp>		Y		
Standard Trailer		Y		

9.2.8 Ticker MarketDataRequest

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType tag 35=V**	
262	MDReqID	Y	A unique ID assigned by the client to the Market Data Request. To unsubscribe from market data, the same ID must be sent with tag 263 = 2.	
263	SubscriptionRequestType	Y	Data request type: T = Trade Ticker (Subscribe) 2 = Unsubscribe	

264	MarketDepth	Y	1 (Only Supported Value)	
265	MDUpdateType	Y	1 = Incremental Refresh	
Component <MdReqGrp>				
267	NoMDEntryTypes	Y	Number of MDEntryType fields being requested. 1 = Only value currently supported	
→	269	MDEntryType	Y	Requested Market Data type: 2 = Trade.	
End Component <MdReqGrp>				
Component <InstrmtMDReqGrp>				
146	NoRelatedSym	Y	Number of related symbols in the request. This value is always equal to '1' in current version.	
→	55	Symbol	Y	Instrument (E.g. BTC/USD)	
End Component <InstrmtMDReqGrp>				
Standard Trailer	Y			

9.2.9 MarketDataRequestReject

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType tag 35=Y**	
262	MDReqID	Y	A unique ID assigned by the client to the Market Data Request. To unsubscribe to market data, the same ID must be sent with tag 263 = 2.	
281	MDReqRejReason	N	Numerical reason for the rejection of the Market Data Request 0=UNKNOWN_SYMBOL 1= DUPLICATE MDReqID	
58	Text	N	Free format text string describing the reason for rejection.	
Standard Trailer		Y		

9.2.10 MarketDataIncrementalRefresh

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType tag 35=X**	
262	MDReqID	Y	A unique ID assigned by the client to the Market Data Request.	
6006	MarketDataID	N	Sequence number which uniquely identifies all unsolicited market data messages within a trade date, for example 35=X and 35=f. Messages containing the same Global Market Data ID within a Trade Date should be considered as duplicates.	V3.10

Component <MDIncGrp>					
268	NoMDEntries		Y	Number of entries following.	
→	279	MDUpdateAction	Y	The Market Data update action type. 0 = New 2 = Delete	
→	285	DeleteReason	N	If MDUpdateAction = Delete (2), this field can be used to specify a reason.	
→	269	MDEntryType	Y	A list of all the Market Data entries types the requesting firm is interested in receiving. 0 = Bid 1 = Offer 4= Opening Price 5= Closing Price 6= Settlement 7= Session High Price 8 = Session Low Price B = Total Volume J= Empty Book	
→	278	MDEntryID	N	Will not be sent for statistics messages (where 269 = 4, 5, 6, 7, 8, B, J) Please refer to Handling MDEntryID in Market Data Messages Section. Changed to a hexadecimal encoding of a long data type as a string.	
→	55	Symbol	N	Instrument (E.g. BTC/USD or BTCZ9)	
→	270	MDEntryPx	N	Price of the corresponding order update.	
→	15	Currency	N	The currency for the amount specified in the MDEntrySize (271) field.	
→	271	MDEntrySize	N		
→	~~110~~	~~MinQty~~	~~N~~	~~The minimum fill size associated with the amount, MDEntrySize (271), and quote, MDEntryPx (270). The FIX id must be enabled and the market data request made as full book non aggregated for this field to be populated.~~	Removed V3.8
→	346	NumberOfOrders	N	Used in an Aggregated Book to show how many	

				individual orders make up an MDEntry	
→	58	Text	N	Text field used to describe the Market Data Entry.	
->	286	OpenCloseSettlFlag	N	5 = Theoretical Price Value. Used to indicate that the price received during pre-open is an indicative price.	
End Component <MDIncGrp>					
→	60	TransactTime	Y	Time of execution in GMT; e.g. YYYYMMDD-HH:MM:SS.000000000 (nanosecond)	
→	6001	EventIndicator	N	2 = EndofEvent. Will be sent at the end of a message sequence to indicate that all prior messages were part of an atomic matching event.	
Standard Trailer			Y		

9.2.11 Ticker MarketDataIncrementalRefresh

Tag	Field Name		Rqd	New Spec	Updated?
Standard Header			Y	**MsgType tag 35=X**	
262	MDReqID		Y	A unique ID assigned by the client to the Market Data Request.	
6006	MarketDataID		N	Sequence number which uniquely identifies all unsolicited market data messages within a trade date, for example 35=X and 35=f. Messages containing the same Global Market Data ID within a Trade Date should be considered as duplicates.	V3.10
Component <NoMDEntries>					
→	268	NoMDEntries	Y	Number of entries	
→	279	MDUpdateAction	Y	0 = New	
→	269	MDEntryType	Y	2 = Trade	
→	55	Symbol	Y	Instrument (E.g. BTC/USD)	
→	270	MDEntryPx	Y	Price	
→	15	Currency	Y	The currency for the amount specified in the MDEntrySize (271) field.	
→	271	MDEntrySize	Y	Trade Quantity	
→	346	NumberOfOrders	Y	Number of orders filled at the price level reported	Added V3.4
→	7562	TickerType	N	ErisX defined tag sent on executions after initial opening trades. G = Given (Seller is Aggressor) P = Paid (Buyer is Aggressor)	
End Component <NoMDEntries>					
→	60	TransactTime	Y	Time of execution in GMT; e.g.	

				YYYYMMDD-HH:MM:SS.000000000 (nanosecond)	
6001	EventIndicator		Y	1 = EndOfTrade. Indicates when no more trades for an event will be published.	
Standard Trailer			Y		

9.2.12 NewOrderSingle

Tag	Field Name	Ord Type			New Spec	Updated?
		1	2	4		
Standard Header		Y	Y	Y	**MsgType tag 35=D**	
11	ClOrdID	Y	Y	Y	Client assigned unique order identifier. Maximum ClOrdID length = 40 characters.	Updated V3.3
Component block <Parties>						
453	NoPartyIDs	N	N	N	Number of PartyIDs	
448	PartyID	N	N	N	Optional. When: 452=3, specify the FIX session on behalf of which the trade is for. 452=24, specify the FCM back office account number	
452	PartyRole	N	N	N	3 = 'On Behalf Off' Routing FIX ID 24 = Customer Account Reference (FCM Back office Account)	
End Component block <Parties>						
581	AccountType	NA	F	F	Required for Futures: 1 = Customer, 2 = House	
582	CustOrderCapacity	NA	F	F	Required for Futures CTICode (customer type indicator) 1 = Member Trading for own account 2 = Clearing firm trading for its Prop Account 3 = Member trading for another member 4 = All other	
21	HandlInst	Y	Y	Y	1 = Automated execution	
18	ExecInst	N	N	N	Only supported value. 6 = Participate don't initiate	Added V3.4
15	Currency	Y	Y	Y	The currency of the OrderQty field	
54	Side	Y	Y	Y	Order side: 1 = Buy, 2 = Sell	
55	Symbol	Y	Y	Y	Instrument (E.g. BTC/USD or BTCZ9)	
460	Product	N	N	Y	2 = Commodity	
60	TransactTime	Y	Y	Y	Request Time: YYYYMMDD-HH:MM:SS.000000000 (nanosecond) Valided to one second precision	
38	OrderQty	N	Y	Y	Order quantity specified in the base Currency. Required for Market orders when Side=Sell.	Updated v3.13

152	CashOrderQty	N	N	N	Order Quantity specified in the quote Currency/ Required for Market orders when Side=Buy.	Added v3.13
40	OrdType	Y	Y	Y	Supported values are: 1 = Market order 2 = Limit order 4 = Stop-Limit order	Updated v3.13
1	Account	N	N	N	Exchange Account Id (Not Required)	
44	Price	NA	Y	Y	Limit or Stop-Limit Price	
99	StopPx	NA	NA	Y	The price at which the stop order becomes effective	
432	ExpireDate	NA	N	N	Expiry date in YYYYMMDD format. Required when TimeInForce = GTD	
59	TimeInForce	Y	N	N	Specifies how long an order remains in effect: 0 = Day 1 = Good Till Cancel 3 = Immediate or Cancel 4 = Fill or Kill 6 = Good Till Date (the ExpireDate (432) If OrdType(40) = 1 (Market order) Then TimeInForce(59)=3	Updated v3.13
110	MinQty	NA	N	N	The minimum quantity for which the order can be executed for TimeInForce (59) = Immediate or Cancel.	
58	Text	N	N	N	Text field	
Standard Trailer		Y	Y	Y		

9.2.13 ExecutionReport: Sent for Order Related Requests

Tag	Field Name	Ord Type			New Spec	Updated?
		1	2	4		
Standard Header		Y	Y	Y	**MsgType tag 35=8**	
37	OrderID	Y	Y	Y	Unique order identifier assigned by ErisX.If 150 = 8 (Rejected), is set to "UNKNOWN."	
11	ClOrdID	Y	Y	Y	Client assigned unique order identifier. Maximum ClOrdID length = 40 characters.	Updated V3.3
41	OrigClOrdID	N	N	N	Original client assigned order id submitted on the order. = 40 characters	Updated V3.3
Component block <Parties>						
453	NoPartyIDs	N	N	N	Number of PartyIDs	
448	PartyID	N	N	N	452=24, specify the FCM back office account number	
452	PartyRole	N	N	N	24 = Customer Account Reference (FCM Back office Account)	

End Component block <Parties>						
17	ExecID	Y	Y	Y	Unique identifier for the execution message assigned by ErisX. Buy orders will begin with '1_' and Sell orders will begin with '2_'	
150	ExecType	Y	Y	Y	The execution report's type. 0 = New 4 = Canceled 5 = Replace 8 = Rejected C = Expired F = Fill Status I = Order Status	
39	OrdStatus	Y	Y	Y	The current state of a chain of orders, same scope as OrderQty, CumQty, LeavesQty, and AvgPx. 0 = New 1 = Partial filled 2 = Filled 4 = Canceled 5 = Replaced 8 = Rejected C = Expired I= Order Status	Updated V3.3
103	OrdRejReason	N	N	N	Only present when ExecType = 8, Rejected. 1 = Unknown Symbol 2 = Exchange Closed 3 = Order exceeds limit 5 = Unknown Order 6 = Duplicate Order (e.g. dupe CLOdID) 11 = Unsupported order characteristic 13 = Incorrect Quantity 15 = Unknown Account 16 = Price exceeds current price band 17 = Invalid Stop Price 18 = Invalid Price 19 = Invalid Order Qty 20 = Currency must match base 21 = Order quantity price product too large 22 = Order may not aggress 23 = Quantity Above Account Asset Balance Limit 25 = Insufficient credit limit 27 = Exceed maximum notional order amount 100 = Instrument Closed 101 = Instrument halted. When OrdRejReason=101, If Instrument is Halted then TAG 58= TRADING HALTED and If Instrument is Paused then TAG 58=TRADING PAUSED	Updated V3.13

					102 = Invalid Cash Order Quantity 103 = Invalid Order Type 104 = Market Order Protection	
1	Account	N	N	N	Exchange Account Id (Not Required)	
581	AccountType	NA	F	F	Required for Futures 1 = Customer 2 = House	
582	CustOrderCapicity	NA	F	F	Required for Futures CTICode (customer type indicator) 1 = Member Trading for own account 2 = Clearing firm trading for its Prop Account 3 = Member trading for another member 4 = All other	
55	Symbol	Y	Y	Y	Instrument (E.g. BTC/USD)	
54	Side		Y	Y	Order side:1 = Buy, 2 = Sell	
38	OrderQty	N	Y	Y	Order quantity specific in the base Currency (15). Not sent if ExecType = 4, Canceled, or 8, Rejected.	
152	CashOrderQty	N	N	N	Order quantity specific in the quote Currency. Not sent if ExecType = 4, Canceled, or 8, Rejected.	Added v3.13
40	OrdType	N	N	N	Supported values are: 2 = Limit order 4 = Stop-Limit order Required for most cases except ExecType = 8	
44	Price	Y	Y	Y	Required for Limit orders	
99	StopPx	NA	N A	Y	Required for Stop orders, OrderType = 4. The price at which the stop order becomes effective.	
15	Currency	N	N	N	The currency for the amount specified in tag 38, OrderQty field.	
59	TimeInForce	Y	Y	Y	Specifies how long an order remains in effect: 0 = Day 1 = Good Till Cancel 3 = Immediate or Cancel 4 = Fill or Kill 6 = Good Till Date (the ExpireDate (432)	

432	ExpireDate	N	N	N	Expiry date in YYYYMMDD format. Required when TimeInForce = GTD	
126	ExpireTime	N	N	N	Time/Date of order expiration (always expressed in UTC)	
18	ExecInst	N	N	N	Only supported value. 6 = Participate don't initiate	Added V3.4
32	LastQty	N	N	N	Quantity bought/sold for this fill. Present when ExecType (150) = F.	
31	LastPx	N	N	N	Price at which the current or last fill was made. Not sent for status requests.	
194	LastSpotRate	N	N	N	Price for the last fill. Not sent for status requests	
151	LeavesQty	Y	Y	Y	Amount of order open for further execution. If the OrdStatus 39 = 4, C, the order is no longer active and LeavesQty can = 0. Otherwise, LeavesQty = OrderQty - CumQty.	
14	CumQty	Y	Y	Y	Total amount of an order currently executed in a chain of partial fills.	
6	AvgPx	N	N	N	The average price at which the order was filled or partially filled.	
75	TradeDate	N	N	N	Trade date. Trades completed after 4 pm CT show the next business day as the trade date.	
60	TransactTime	N	N	N	Time of execution in GMT; e.g. YYYYMMDD-HH:MM:SS.000000000 (nanosecond)	
5001	UnsolicitedCancel	N	N	N	Used to indicate the reason for an unsolicited cancel 1 = Self Match Prevention 2 = Message Limits Exceeded 3 = Cancelled due to disconnect 4 = End of Trading Session 5 = Exchange Cancelled 6 = Order May Not Aggress	
Component <CommissionData>						
12	Commission	N	N	N	Actual Commission (Only for Fills and Partial Fills)	
7012	CommCalculated	Y	Y	Y	Calculated Commission	
13	CommType	Y	Y	Y	3 = Absolute (Total monetary amount)	
479	CommCurrency	Y	Y	Y	Currency Commission (USD, BTC)	
End Component <CommissionData>						
110	MinQty	N	N	N	The Minimum quantity for which the order can be executed for TimeInForce (59) = Immediate or Cancel	

58	Text	N	N	N	Descriptive text message.	
7585	MatchingType	N	N	N	Valid values: 1 = Order initiator is aggressor 2 = Order initiator is passive	
Component <PurchasingPowerData>						
5002	NoOfPurchasing Power	N	N	N	Number of purchasing power repeating groups	Added V3.9
5003	PurchasingPow erCurrency	N	N	N	Asset type of the collateral available for trading	Added V3.9
5004	PurchasingPow er	N	N	N	Collateral available for trading	Added V3.9
End Component <AvailableBalanceData>						
Standard Trailer		Y	Y			

9.2.14 OrderReplace

Tag	Field Name	Ord Type		New Spec	Updated?
		2	4		
Standard Header		Y	Y	**MsgType tag 35=G**	
37	OrderID	Y	Y	The ErisX assigned ID of the order to be replaced.	
41	OrigClOrdID	Y	Y	The unique client ID assigned to the order to be replaced.	
11	ClOrdID	Y	Y	Unique client id for the replacement order. Note that this identifier will be used in ClOrdID field of the Cancel Reject message if the replacement request is rejected. Maximum ClOrdID length = 40 characters.	Updated V3.3
1	Account	N	N	Exchange Account Id (Not Required)	
581	AccountType	F	F	Required for Futures 1 = Customer, 2 = House	
582	CustOrderCapici ty	F	F	Required for Futures CTICode (customer type indicator) 1 = Member Trading for own account 2 = Clearing firm trading for its Prop Account 3 = Member trading for another member 4 = All other	
21	HandlInst	Y	Y	Instructions for how order is to be handled by ErisX.	
55	Symbol	Y	Y	Instrument (E.g. BTC/USD) Must match original order.	
460	Product	N	N	2 = Commodity	
54	Side	Y	Y	Order side: 1 = Buy, 2 = Sell Must match side specified in original order.	

45

18	ExecInst	N	N	Only supported value. 6 = Participate don't initiate	Added V3.4
60	TransactTime	Y	Y	Request Time: YYYYMMDD-HH:MM:SS.000000000 (nanosecond) Valided to one second precision	
38	OrderQty	N	N	Order quantity specific in the base Currency (15).	
40	OrdType	Y	Y	The following order types can be replaced: 2 = Limit Order 4 = Stop Limit Order	
44	Price	Y	Y	Limit or Stop-Limit Price.	
99	StopPx	NA	Y	The price at which the stop order becomes effective.	
15	Currency	N	N	The currency for the amount specified in the OrderQty (38) field.	
432	ExpireDate	N	N	Can be specified if order was submitted with TimeInForce (59) = GTD Date for GTD can only be set to 100 days in the future.	
59	TimeInForce	N	N	Specifies how long the order remains in effect. 0 = Day(session) 1= Good Till Cancel(GTC) 6 = Good Till Date(GTD)	Corrected V 3.6
110	MinQty	N	N	The minimum quantity that the order can be executed for TimeInForce (59) = Immediate or Cancel.	
58	Text	N	N	Descriptive text message.	
5000	Overfill Protection	N	N	Required when trying to modify a partially filled order to specifically request "Overfill Protection" otherwise the modification is rejected. Y = LeavesQty is set to requested quantity - CumQty N = LeavesQty is set to the quantity requested in the cancel replace message	
Standard Trailer		Y	Y		

9.2.15 OrderCancelRequest

Tag	Field Name	Ord Type		New Spec	Updated?
		2	4		
Standard Header		Y	Y	MsgType tag 35=F	
11	ClOrdID	Y	Y	The client assigned unique ID for this cancel request. Set to "OPEN_ORDER" to cancel all open orders for this client. Maximum ClOrdID length = 40 characters.	Updated V3.3
41	OrigClOrdID	Y	Y	The client assigned ID of the order to be canceled. Set to "OPEN_ORDER" to cancel all open orders for this client.	

Tag	Field Name			New Spec	Updated?
37	OrderID	Y	Y	The ErisX assigned ID of the order to be canceled. Set to "OPEN_ORDER" when 11 & 41 = "OPEN_ORDER" as its a mandatory field, so needs to be present but the cancel all operation does not look at the content.	
Component block <Parties>				REQUIRED WHEN CANCELLING ALL ORDERS ON BEHALF OF ANOTHER FIX SESSION:	
453	NoPartyIDs	N	N	Number of PartyIDs = 1	Added V3.8
448	PartyID	N	N	452=3, specify the FIX session on behalf of which the cancel request is for.	Added V3.8
452	PartyRole	N	N	3 = 'On Behalf Off' Routing FIX ID	Added V3.8
End Component block <Parties>					
1	Account	N	N	Exchange Account Id (Not Required)	
55	Symbol	Y	Y	Instrument (E.g. BTC/USD). Note for canceling all open orders: Set to 'NA'	
460	Product	N	N	2 = Commodity	
54	Side	Y	Y	Side of order: 1 = Buy, 2 = Sell	
60	TransactTime	Y	Y	Request Time: YYYYMMDD-HH:MM:SS.000000000 (nanosecond) Valided to one second precision	
58	Text	N	N	Descriptive text message.	
40	OrdType	NA	Y	Order type. Must be present on cancel requests made for the following non primitive order types:4 = Stop-Limit order	
7559	OpenOrders	N	N	Y = Cancel all open orders. Required when tags 11 and 37 = "OPEN_ORDER."	
Standard Trailer		Y	Y		

9.2.16 OrderCancelReject

Tag	Field Name	Ord Type		New Spec	Updated?
		2	4		
Standard Header		Y	Y	**MsgType tag 35=9**	
11	ClOrdID	Y	Y	The client assigned unique ID for the cancel request being rejected.	
41	OrigClOrdID	Y	Y	ClOrdID for the order that could not be canceled or replaced.	
37	OrderID	Y	Y	The ErisX ID for the order that could not be canceled or replaced. If the order id cannot be determined, i.e., CxlRejReason = "Unknown order" or if the order is not active, "NONE" will be specified.	
39	OrdStatus	Y	Y	8 = Rejected	

47

				Note: The treatment of this Tag is non-standard. The value is the status of the Order Cancel Request, not of any order and should not be processed.	
Component block <Parties>				Only included when sen in the original request	Added V3.8
453	NoPartyIDs	N	N	Number of PartyIDs = 1	Added V3.8
448	PartyID	N	N	452=3, specify the FIX session on behalf of which the cancel request is for.	Added V3.8
452	PartyRole	N	N	3 = 'On Behalf Off' Routing FIX ID	Added V3.8
End Component block <Parties>					
1	Account	N	N	Exchange Account Id (Not Required)	
60	TransactTime	Y	Y	Time in GMT; e.g. YYYYMMDD-HH:MM:SS.000000000 (nanosecond)	
434	CxlRejResponseTo	Y	Y	Specifies to what the reject is in response: 1 = Order Cancel Request 2 = Order Replace Request	
102	CxlRejReason	N	N	Reason the order cancellation request was rejected: 0 = Order has already been filled 1 = Order cannot be found 2 = Check tag 58 for details 3 = Status cannot be determined as order is currently in process. Execution Report returned, once processing completes will contain status. 14 = Quantity Above Account Asset Balance Limit 17 = Invalid Order Type	Updated V3.13
58	Text	N	N	Descriptive text message: "Cancel Non Active" = Order is not active "clOrdId already exists" = Duplicate clOrderId has been received "Cancel Failed I O C" = Cannot cancel an IOC order "No Resting Orders" = No orders working at the time of the request **"TRADING PAUSED"=Only Cancel messages are permitted** **"TRADING HALTED"=Order Add/Cancel/Modify are NOT Permitted**	Updated V3.12
Standard Trailer		Y	Y		

9.2.17 Order Mass Status Request

Tag	Field Name	Rqd	New Spec	Updated?
Standard Header		Y	**MsgType tag 35=AF**	
584	MassStatusReqID		Y	Mass status unique request ID	

585	MassStatusReqType	Y	8 = Status of all orders related to session party	
60	TransactTime	N	Request Time: YYYYMMDD-HH:MM:SS.000000000 (nanosecond) Valided to one second precision	
Component block <Parties>				
453	NoPartyIDs	N	Number of PartyIDs	
448	PartyID	N	Optional. When: 452=3, specify the FIX session on behalf of which the trade is for. 452=24, specify the FCM back office account number	
452	PartyRole	N	3 = 'On Behalf Off' Routing FIX ID 24 = Customer Account Reference (FCM Back office Account)	
Standard Trailer		Y		

9.2.18 Execution Report (response sent to the OrderMassStatusRequest)

Tag	Field Name	Ord Type		New Spec	Updated
		2	**4**		
Standard Header				MsgType tag 35=8	
37	OrderID	Y	Y	Unique order identifier assigned by ErisX. If 911=0 This value will be either 0 or NA	
11	ClOrdID	Y	Y	Client assigned order id to the current order action. If 911=0 This value will be either 0 or NA	
41	OrigClOrdID	Y	Y	Original client assigned order id. If 911=0 This value will be either 0 or NA	
57	TargetSubID	F	F	Value used to identify the user that entered the order	Added V3.13
143	TargetLocationID	F	F	Used to identify the geographical location of the user that entered the order: [Country],[State if in US] eg; US, IL or UK	Added V3.13
584	MassStatusReqID	N	N	Mass status unique request ID	
911	TotNumReports	N	N	Total no of execution reports sent for that order. 'O' is sent if no execution reports exist.	
912	LastRptRequested	N	N		
Component <Parties>					
453	NoPartyIDs	Y	Y	Number of parties.	
448	PartyID	Y	Y	The specified value based upon the PartyRole (452)	
452	PartyRole	Y	Y	Supported Values: 1 = ExecutingFirm 3 = ClientID (Eris Digital account userID for who the order belongs to)	

				11= Order Origination Trader (associated with Order Origination Firm e.g. trader who initiates/submits the order) 24 = Customer Account Reference	
End Component <Parties>					
17	ExecID	Y	Y	ExecID for responses to OrderMassStatusRequest will be 0	
150	ExecType	Y	Y	The execution report's type. Contains one more value than tag 39, OrderStatus I = Order Status	
39	OrdStatus	Y	Y	The current state of chain of orders, e.g., when there are partial fills. Has the same scope as OrderQty, CumQty, LeavesQty, and AvgPx. 0 = New, 1 = Partial	
1	Account	N	N	Exchange Account Id (Not Required)	
581	AccountType	F	F	Required for Futures: 1 = Customer, 2 = House	
582	CustOrderCapicity	F	F	Required for Futures: CTICode (customer type indicator) 1 = Member Trading for own account 2 = Clearing firm trading for its Prop Account 3 = Member trading for another member 4 = All other	
55	Symbol	Y	Y	Instrument (E.g. BTC/USD)	
54	Side	Y	Y	1 = Buy, 2 = Sell	
38	OrderQty	Y	Y	The order amount in the currency specified in tag 15	
40	OrdType	N	N	2 = Limit order, 4 = Stop-Limit order	
44	Price	Y	Y	Required for Limit based orders. Price at which the limit order is to be executed.	
99	StopPx	NA	Y	Required for Stop orders, OrderType = 4. The Price at which the stop order becomes effective.	
15	Currency	N	N	The dealt currency of the order. This is the currency for the amount specified in tag 38, OrderQty field.	
32	LastQty	N	N	Quantity bought/sold for a partially filled order.	
31	LastPx	N	N	Price at which the current or last fill was made	
151	LeavesQty	Y	Y	LeavesQty (151) = OrderQty (38) − CumQty (14).	
14	CumQty	Y	Y	Total amount of an order currently executed in a chain of partial fills.	
6	AvgPx	N	N	The average price at which the order was filled or partially filled	
59	TimeinForce	Y	Y	0=Day 1=Good Till Cancel	

				3=Immediate of Cancel 4=Fill or Kill 6=Good Till Date	
Component <CommissionData>					
12	Commission	N	N	Actual Commission (For Partial Fills)	Added V3.8
701 2	CommCalculated	Y	Y	Calculated Commission	Added V3.8
13	CommType	Y	Y	3 = Absolute (Total monetary amount)	Added V3.8
479	CommCurrency	Y	Y	Currency Commission (USD, BTC)	Added V3.8
End Component <CommissionData>					
60	TransactTime	N	N	Time this order request was initiated or released by the trader or trading system. Execution Reports will be sent with nanosecond precision - YYYYMMDD-HH:MM:SS.000000000	
Component <PurchasingPowerData>					
500 2	NoOfPurchasingP ower	N	N	Number of purchasing power repeating groups	Added V3.9
500 3	PurchasingPower Currency	N	N	Asset type of the collateral available for trading	Added V3.9
500 4	PurchasingPower	N	N	Collateral available for trading	Added V3.9

Appendices

A. Coordinated Universal Time (UTC) Format

All time and date formats must be in Coordinated Universal Time (UTC).

- Year: YYYY (2003)
- Year and month: YYYYMM (200307)
- Year, Month, Day: YYYYMMDD (20030716)
    - plus hours, minutes and seconds: YYYYMMDD-hh:mm:ss (20030716-19:20:30)
    - plus hours, minutes, seconds and milliseconds YYYYMMDD-hh:mm:ss.mmm (20030716-19:20:30.183)
- Transact Time hours, minutes, seconds, and nanoseconds: YYYYMMDD-HH:MM:ss.SSSSSSSSS (20180716- 23:15:56.612339123)

where:

Format	Description
YYYY	four digit year
MM	two digit month (01=January, etc.)
DD	two digit day (01 through 31)
hh	two digit hour (00 through 23) (am/pm NOT allowed)
mm	two digit minute (00 through 59)
ss	two digit second (00 through 59)
mmm	three digit millisecond (000 – 999)
SSSSSS	six digit microsecond (000000 - 999999)
SSSSSSSSS	9 digit nanoseconds (000000000 - 999999999)