ErisX

FIX 4.4 Specification V2.2
Market Data and Order Entry

ErisX

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

Contents

ErisX

ErisX

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.
	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
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.

ErisX

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.

3.2 Supported Time in Force

The supported time in force values are 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 day 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 cancelled.
Immediate or Cancel (IOC)	Orders with the expiry condition will be cancelled unless a specified minimum quantity can be executed immediately at the specified price or better. Any remaining unfilled quantity is cancelled.

3.3 Order Modification

Order parameters such as quantity and expiry 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 user makes use of the overfill protection logic. A reject message will be received if attempting to modify a partially filled order without the use of overfill protection.

ErisX

3.4 Minimum Permitted Order Entry Size

There is a minimum permitted order entry size maintained on 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 finger" 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 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.

ErisX

4 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 need to be guaranteed.

4.1 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) included, 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.

4.2 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.

ErisX

- 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.

4.3 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.

4.4 Application Messages

Once a proper session is established, application messages are used to receive market data, to submit orders, and to receive executions 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 instruments requests are not supported at this time.

ErisX

- **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. 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 order history by sending a MassOrderStatusRequest

ErisX

5 Message Workflow

5.1 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.

5.2 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
- 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 instruments market data.

5.3 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.

ErisX

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 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.

ErisX

6 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 Custom tag EventIndicator (6001=1) representing 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)
```

6.1 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
- TopOfBook – 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.

6.2 FullBook Aggregated Incremental

- 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.

6.3 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.

ErisX

6.4 TopOfBook Aggregated Incremental

- Only the best prices for each instrument, with amounts aggregated are sent to the client.
- Cancels are only delivered if the best price is affected in the instrument.
- Updates are only delivered to the client if the best price for an instrument is affected. This could be a new best price or a new aggregate size

6.5 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.

Within the same pair, only one (1) price can be outstanding for any one 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.

6.6 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.

6.7 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.

ErisX

7 Order Processing Messages

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

7.1 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 filled, modified or canceled.

Client	Direction	Message info	ErisX
Submit New Order			
Submit order	→	OrderSingle	
	←	ExecutionReport (150=0; 39=0)	Respond: confirm order or reject order
Order Fill			
	←	ExecutionReport (150=F; 39=1 or 2)	Send order partial or full fill notification
Order Fill is Timed Out			
	←	ExecutionReport (150=9)	Send order fill timeout notification
Request to Cancel Active Order			
Request order cancel	→	OrderCancelRequest	
	←	ExecutionReport (150=4; 39=4) - If order is canceled or OrderCancelReject - If cancel request failed	Respond: order is canceled or cancel request has failed
Order Replace Request			
Request order replace	→	OrderCancelReplaceRequest	
	←	ExecutionReport (150=5; 39=5) – If order is replaced or OrderCancelReject – If replace request failed	Respond: order is canceled or cancel request has failed
Order is Canceled			
	←	ExecutionReport (150=4; 39=4)	Send order canceled notification
Order is Expired			
	←	ExecutionReport (150=C; 39=C)	Send order expired notification
Order Mass Status Request			
Request working orders	→	OrderMassStatusRequest (60= YYYYMMDD-HH:MM:SS)	
	←	ExecutionReport (150=I) for that order will be sent until 912=Y	Respond with execution reports

ErisX

7.2 Order Expiry Types

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

Note: Previous version 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)

7.3 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.

7.4 Handling Fill Type Execution Reports

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

ErisX

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.

7.5 Account and ClientID

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

On an NewOrderSingle message, the Account (1) is used to identify the booking entity. ErisX defaults the booking id to be the same as the SenderCompId (49) used to establish the FIX connection and the Account (1) to be the ErisX defined entity under which the SenderCompID exists.

A client with multiple trading entities can be configured to trade on behalf of each entity via a single FIX connection. The appropriate entity is indicated on the OrderSingle message by setting Party (448) to an id defined under the target entity and PartyRole (452) to '3' for on behalf of client. The trade, if done, will then be booked under the corresponding id and entity.

The Account and ClientID fields 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.

7.6 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 users the overfill protection functionality tag 5000

ErisX

7.7 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 LeavesQty = Replace message Qty - cumQty
- Whereas with Overfill Protection = N, the remaining quantity (LeavesQty) is set to the quantity was 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: This an original order to buy 5 lots which have been partially filled

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

When: A modify request is received containing OrderQty(38) of 4 with Overfill Protection = **Y**
Then: The order quantity is set to 4, which reduces the LeavesQty quantity down

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

Or:

When: A modify request is received containing OrderQty(38) of 4 with Overfill Protection = N
Then: The remaining quantity is set to 4

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

7.8 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

7.9 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]”

ErisX

7.10 Mass Order Status Request

The mass order status request should be used to obtain a set of working orders for a given session. Previous versions of this request were responded to with a full set of order history. This has been removed in our new match engine.

Members who are disconnected should use the resend request message logic to receive any missed messages.

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

7.11 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.

7.12 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 two independent message throttle limits and a configurable time interval:
    - Max messages per X seconds
    - Max orders of operations per X seconds
        - Order Operations include New Orders, Order Mass Status Request, and Modify
- 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 configurable time interval, the message counter is reset.
    - This is not a rolling interval

If a session breeches 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.

ErisX

8 Message Details

The fields that make up each message are described in this section.

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 message 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.

8.1 Session Messages

8.1.1 Standard Header

Tag	Field Name	Rqd	New Spec	What's Changed?
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.	
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.	

ErisX

8.1.2 Standard Trailer

Tag	Field Name	Rqd	New Spec	What's Changed?
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.	

8.1.3 Logon

Tag	Field Name	Rqd	Comments	What's Changed?
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	Id specific password.	
Standard Trailer		Y		

8.1.4 Logout

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

8.1.5 Resend Request

Tag	Field Name	Rqd	New Spec	What's Changed?
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. If request is for all messages subsequent to a particular message, EndSeqNo = 0.	
Standard Trailer		Y		

ErisX

8.1.6 Test Request

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

8.1.7 Heartbeat

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

8.1.8 User Request (Change Password)

Tag	Field Name	Rqd	New Spec	What's Changed?
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		

8.1.9 User Response

Tag	Field Name	Rqd	New Spec	Whats Changed?
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		

ErisX

8.2 Application Messages

8.2.1 TradingSessionStatus

Tag	Field Name	Rqd	New Spec	What's Changed?
Standard Header		Y	MsgType tag 35=h	
336	TradingSessionID	Y	Identifier for this trading session.	
340	TradSesStatus	Y	101=System Ready 105=System Disconnect	State of the trading session: ~~2 = Open Session~~ ~~5 = Pre-Close Session~~ 101=System Ready ~~102 = Open Trading~~ ~~103 = Closed Trading~~ ~~104 = Pre-open Trading~~ 105=System Disconnect
342	TradSesOpenTime	N	~~Time when the trading will be enabled,~~ ~~present when 340 = 104~~	Removed
344	TradSesCloseTime	N	~~Time when the trading will be disabled,~~ ~~present when 340 = 105~~	Removed
58	Text	N	Descriptive text message.	
Standard Trailer		Y		

8.2.2 BusinessMessageReject

Tag	Field Name	Rqd	New Spec	What's Changed?
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		

8.2.3 SecurityListRequest

Tag	Field Name	Rqd	New Spec	What's Changed?
Standard Header		Y	MsgType tag 35=x (lowercase)	
320	SecurityReqID	Y	Unique security request ID.	
559	SecurityListRequestType	Y	0 = Symbol	
Component <Instrument>		Y		
55	Symbol	Y	Set to NA	
460	Product	Y	2 = Commodity	
End Component <Instrument		Y		
Standard Trailer		Y		

8.2.4 SecurityList

Tag	Field Name		Rqd	New Spec	What's changed?
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>		Y		
→	55	Symbol	Y	Instrument (E.g. BTC/USD)	
→	~~65~~	~~SymbolSfx~~	~~Y~~	~~SP = spot (default if not specified)~~	Removed
→	460	Product	Y	2 = Commodity	
→	969	MinPriceIncrement	Y		
→	107	SecurityDesc	Y		
→	562	MinTradeVol	N	The minimum order quantity that can be submitted for an order.	

ErisX

→	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	
	End Component <Instrument>		Y		
End Component <SecListGrp>			Y		
Standard Trailer			Y		

8.2.5 SecurityStatus

Tag	Field Name	Rqd	New Spec	What's Changed?
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)	
~~65~~	~~SymbolSfx~~	~~Y~~	~~SP = spot (default if not specified)~~	Removed
460	Product	Y	2 = Commodity	
969	MinPriceIncrement	Y		
107	SecurityDesc	Y	E.g. Bitcoin USD	
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.	
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	Added 2, 15, 21, 28 17 = Ready (Updated) 18 = Not Available (Updated)
1174	SessionEnd	N	4 = Change of Trading Session	
Standard Trailer		Y		

8.2.6 MarketDataRequest

Tag	Field Name	Rqd	New Spec	What's Changed?

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	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 1 = Top of 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>					
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)	
→	~~65~~	~~SymbolSfx~~	~~N~~		Removed
End Component <InstrmtMDReqGrp>					
Standard Trailer			Y		

8.2.7 Ticker MarketDataRequest

Tag	Field Name	Rqd	New Spec	What's Changed?
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.	

ErisX

263	SubscriptionRequestType		Y	Data request type: T = Trade Ticker (Subscribe) 2 = Unsubscribe	
264	MarketDepth		Y	1 = Top of Book	
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)	
→	~~65~~	~~SymbolSfx~~	~~N~~	~~SP = spot (default if not specified)~~	Removed
End Component <InstrmtMDReqGrp>					
Standard Trailer			Y		

8.2.8 MarketDataRequestReject

Tag	Field Name	Rqd	New Spec	What's Changed?
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	
58	Text	N	Free format text string describing the reason for rejection.	
Standard Trailer		Y		

8.2.9 MarketDataIncremental

Tag	Field Name	Rqd	New Spec	What's Changed?
Standard Header		Y	**MsgType tag 35=X**	
262	MDReqID	Y	A unique ID assigned by the client to the Market Data Request.	
Component <MDIncGrp>				
268	NoMDEntries	Y	Number of entries following.	

ErisX

→	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	Added 4, 5, 6, 7, 8, B, J
→	278	MDEntryID	Y	Market data identifier. There is only one MDEntry active at any one time. MDEntryID (278) is unique per instrument and session. A market data update message with an MDUpdateAction = 0 (New) should replace any active entry with the same MDEntryID (278). A market data update message with MDUpdateAction (279) = 2 (Delete) indicates any active entry with the same MDEntryID should be deleted.	
→	55	Symbol	Y	Instrument (E.g. BTC/USD)	
~~→~~	~~65~~	~~SymbolSfx~~	~~N~~	~~SP = spot (default if not specified)~~	Removed
→	270	MDEntryPx	N	Price.	
→	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.	
→	346	NumberOfOrders	N	Used in an Aggregated Book to show how many individual orders make up an MDEntry	
→	15	Currency	N	The currency for the amount specified in the MDEntrySize (271) field.	
→	58	Text	N	Text field used to describe the Market Data Entry.	

→	60	TransactTime	Y	Time of execution in GMT; e.g. YYYYMMDD-HH:MM:SS.000000000 (nanosecond)	Nanosecond precision
->	286	OpenCloseSettlFlag	N	5 = Theoretical Price Value. Used to indicate that the price received during pre-open is an indicative price.	Added
End Component <MDIncGrp>					
→	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.	Added
Standard Trailer			Y		

8.2.10 Ticker MarketDataIncrementalRefresh

Tag	Field Name	Rqd	New Spec	What's Changed?
Standard Header		Y		
262	MDReqID	Y	A unique ID assigned by the client to the Market Data Request.	
Component <NoMDEntries>				
→	268	NoMDEntries	Y	Number of entries	Multiple Entries now supported
→	279	MDUpdateAction	Y	0 = New	
→	269	MDEntryType	Y	2 = Trade	
→	55	Symbol	Y	Instrument (E.g. BTC/USD)	
~~→~~	~~65~~	~~SymbolSfx~~	~~N~~	~~SP = spot (default if not specified)~~	Removed
→	270	MDEntryPx	Y	Price	
→	15	Currency	Y	The currency for the amount specified in the MDEntrySize (271) field.	
→	271	MDEntrySize	Y	Trade Quantity	
→	600	TickerType	N	ErisX defined tag sent on executions after intial opening trades. G = Given (Aggressive) P = Paid (Passive)	
→	60	TransactTime	Y	Time of execution in GMT; e.g. YYYYMMDD-HH:MM:SS.000000000 (nanosecond)	Nanosecond precision
End Component <NoMDEntries>					

ErisX

6001	EventIndicator	Y	1 = EndOfTrade. Indicates when no more trades for an event will be published.	New Tag
Standard Trailer		Y		

8.2.11 NewOrderSingle

Tag	Field Name	Ord Type		New Spec	What's Changed?
		2	**4**		
Standard Header		Y	Y		
11	ClOrdID	Y	Y	Client assigned unique order identifier. Maximum ClOrdID length = 50 characters.	Max length changed to 50
Component block <Parties>					
453	NoPartyIDs	N	N	Number of PartyIDs	
448	PartyID	N	N	Optional. Populated with PartyRole (452) = 3, specify the trading user on behalf of which the trading is done.	
452	PartyRole	N	N	3 = ClientID; Used if trading on behalf of 'Client ID' for FIX 4.4 only.	
End Component block <Parties>					
21	HandlInst	Y	Y	1 = Automated execution	
15	Currency	Y	Y	The currency for the amount specified in the OrderQty (38) field.	
54	Side	Y	Y	Order side: 1 = Buy, 2 = Sell	
55	Symbol	Y	Y	Instrument (E.g. BTC/USD)	
~~65~~	~~SymbolSfx~~	~~N~~	~~N~~	~~SP = spot (default if not specified)~~	Removed
460	Product	Y	Y	2 = Commodity	
60	TransactTime	Y	Y	Request Time: YYYYMMDD-HH:MM:SS.000000000 (nanosecond) Valided to one second precision	Nanosecond precision
38	OrderQty	Y	Y	Order quantity specific in the base Currency (15).	
40	OrdType	Y	Y	Supported values are: 2 = Limit order 4 = Stop-Limit order	
1	Account	N	N	The account (SubFund id) name as agreed to by ErisX and the client or else defaulted by the system."A FIX connection(i.e. a session), can place orders on behalf of many different accounts"	
44	Price	Y	Y	Limit or Stop-Limit Price.	

ErisX

99	StopPx	NA	Y	The price at which the stop order becomes effective.	
432	ExpireDate	N	N	Expiry date in YYYYMMDD format. Required when TimeInForce = GTD	
59	TimeInForce	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) TimeInForce (59) settings of Immediate or Cancel (59=3,) and Fill or Kill (59=4) are not supported on any complex order types: 4: Stop-Limit order	Add Values: 3 = Immediate or Cancel 4 = Fill or Kill
110	MinQty	N	N	The minimum quantity for which the order can be executed for TimeInForce (59) = Immediate or Cancel.	
58	Text	N	N	Text field	
~~7534~~	~~StopSide~~	~~NA~~	~~Y~~	~~Valid values:~~ ~~1 = Bid~~ ~~2 = Offer~~ ~~Mandatory field for OrdTypes: 40 = 4.~~	Removed
Standard Trailer		Y	Y		

8.2.12 ExecutionReport: Sent for Order Related Requests

Tag	Field Name	Ord Type		New Spec	What's Changed?
		2	**4**		
Standard Header		Y	Y		
37	OrderID	Y	Y	Unique order identifier assigned by ErisX.If 150 = 8 (Rejected), is set to “UNKNOWN.”Note: this is not the same as the completed trade id, which appears in tag 17, ExecID.	
11	ClOrdID	Y	Y	Client assigned unique order identifier. Maximum ClOrdID length = 50 characters.	Max length changed to 50
41	OrigClOrdID	N	N	Original client assigned order id submitted on the order. = 50 characters	Max length changed to 50
17	ExecID	Y	Y	Completed trade identifying number:If tag 150 = 0 and 39 = 0, trade has not been completed yet, shows an internally generated Cx number, e.g., 10813_76982. For a chain of partial fills will contain a draft order id.If tags 150 = F and 39 = 2, shows the final executed trade id. If 150 = 8 (Rejected), is set to “UNKNOWN.”	

ErisX

150	ExecType	Y	Y	The execution report's type. Contains one more value than tag 39 OrderStatus: 0 = New 4 = Canceled 5 = Replace 8 = Rejected C = Expired F = Fill Status 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 filled 2 = Filled 4 = Canceled 5 = Replaced 8 = Rejected C = Expired	
103	OrdRejReason	N	N	Optional with 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 18 = Invalid Price Increment 25 = Insufficient credit limit 27 = Exceed maximum notional order amount 101 = Instrument halted	Values were not listed in previous spec
1	Account	N	N	The account (SubFund id) name as agreed to by ErisX and the client or else defaulted by the system. A FIX connection can have one or more trading accounts.	
55	Symbol	Y	Y	Instrument (E.g. BTC/USD)	
~~65~~	~~SymbolSfx~~	~~N~~	~~N~~		Removed
54	Side	Y	Y	Order side:1 = Buy, 2 = Sell	
38	OrderQty	Y	Y	Order quantity specific in the base Currency (15). Not sent if ExecType = 4, Canceled, or 8, Rejected.	
40	OrdType	N	N	Supported values are: 2 = Limit order 4 = Stop-Limit order Required for most cases except ExecType = 8	

ErisX

44	Price	Y	Y	Required for Limit orders	
99	StopPx	NA	Y	Required for Stop orders, OrderType = 4. The price at which the stop order becomes effective.	
15	Currency	N	N	The currency for the amount specified in tag 38, OrderQty field.	
59	TimeInForce	Y	Y	How long an order remains in effect: 1 = Good Till Cancel 6 = Good Till Date (the ExpireDate (432)	
432	ExpireDate	Y	Y	How long an order remains in effect: 0 = Day 1 = Good Till Cancel 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK) 6 = Good Till Date	Added new fields for IOC and FoK
126	ExpireTime	Y	Y	Time/Date of order expiration (always expressed in UTC)	
32	LastQty	N	N	Quantity bought/sold for this fill. Present when ExecType (150) = F.	
31	LastPx	N	N	Price at which the current or last fill was made. Not sent for status requests.	
194	LastSpotRate	N	N	Price for the last fill. Not sent for status requests	
151	LeavesQty	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	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.	
75	TradeDate	N	N	Trade date. Trades completed after 4 pm CT show the next business day as the trade date.	
60	TransactTime	N	N	Time of execution in GMT; e.g. YYYYMMDD-HH:MM:SS.000000000 (nanosecond)	Nanosecond precision
Component <CommissionData>					
12	Commission	N	N	Actual Commission (Only for Fills and Partial Fills)	
7012	CommCalculated	Y	Y	Calculated Commission	
13	CommType	Y	Y	3 = Absolute (Total monetary amount)	
479	CommCurrency	Y	Y	Currency Commission (USD, BTC)	
End Component <CommissionData>					
110	MinQty	N	N	The Minimum quantity for which the order can be executed for TimeInForce (59) = Immediate or Cancel	Added for IOC orders

ErisX

58	Text	N	N	Descriptive text message.	
~~7530~~	~~BelowMin~~	~~N~~	~~N~~		Removed
~~7534~~	~~StopSide~~	~~NA~~	~~N~~		Removed
7585	MatchingType	N	N	Indicates whether or not the maker's price was resting in the book at the time of the match. Valid values: 1 = Taker is the aggressor 2 = Taker is not the aggressor NOTE: this tag is only supported for Fill (Trade) messages.	
Standard Trailer		Y	Y		

8.2.13 OrderReplace

Tag	Field Name	Ord Type		New Spec	What's Changed?
		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 = 50 characters.	Max length changed to 50
1	Account	N	N	The account (Subfund) name as agreed to by ErisX and the client or else defaulted by the system. A FIX connection can have one or more trading accounts.	
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.	
~~65~~	~~SymbolSfx~~	~~N~~	~~N~~		Removed
460	Product	N	N	2 = Commodity	
54	Side	Y	Y	Order side: 1 = Buy 2 = Sell Must match side specified in original order.	
60	TransactTime	Y	Y	Request Time: YYYYMMDD-HH:MM:SS.000000000 (nanosecond) Valided to one second precision	Nanosecond 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.	
110	MinQty	N	N	The minimum quantity for which the order can be executed for TimeInForce (59) = Immediate or Cancel.	Added
58	Text	N	N	Descriptive text message.	
5000	Overfill Protection	Y	Y	Required to modify a partially filled order to specifically request "Overfill Protection" otherwise the modification is rejected. Takes Value Y or N If Value=Y LeavesQty = Replace message Qty - Filled Qty	Added
Standard Trailer		Y	Y		

8.2.14 OrderCancelRequest

Tag	Field Name	Ord Type		New Spec	What's Changed?
		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 = 50 characters.	Max length changed to 50
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.	
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.	
1	Account	N	N	The account (SubFund id) name as agreed to by ErisX and the client or else defaulted by the system.	
55	Symbol	Y	Y	Instrument (E.g. BTC/USD). Note for canceling all open orders: Set to ‘NA’	
~~65~~	~~SymbolSfx~~	~~N~~	~~N~~		Removed
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	Nanosecond precision
58	Text	N	N	Descriptive text message.	

ErisX

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		

8.2.15 OrderCancelReject

Tag	Field Name	Ord Type		New Spec	What's Changed?
		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.	Max length changed to 50
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 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.	
1	Account	N	N	The account (SubFund id) name as agreed to by ErisX and the client or else defaulted by the system. "Note: (Tag 1) Account can be present on CancelReject messages if the CancelRequest message is received whilst an order is being processed. If the order is in the process of being filled, the CancelReject message will subsequently be sent"	
60	TransactTime	Y	Y	Time in GMT; e.g. YYYYMMDD-HH:MM:SS.000000000 (nanosecond)	Nanosecond Precision
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.	
58	Text	N	N	Descriptive text message: “co Cancel Non Active” = Order is not active “clOrdId already exists” = Duplicate clOrderId has been received “co Cancel Failed I O C” = Cannot cancel an IOC order List=instruments, if any, not associated with an active book.	Added “co Cancel Failed I O C” = Cannot cancel an IOC order
Standard Trailer		Y	Y		

8.2.16 Order Mass Status Request

Tag	Field Name	Rqd	New Spec	What's Changed?
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	No Longer Required
Standard Trailer		Y		

8.2.17 Execution Report (response sent to the OrderMassStatusRequest)

Tag	Field Name	Ord Type		New Spec	What's Changed?
		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	Removed: IF 150=8 (Rejected), is set to “UNKNOWN.”
~~526~~	~~SecondaryClOrdID~~	~~N~~	~~N~~		Removed
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 submitted on the order. If 911=0 This value will be either 0 or NA	
584	MassStatusReqID	N	N	Mass status unique request ID	
911	TotNumReports	N	N	Total no of execution reports sent for that order. If no execution reports present this value will be 0	
912	LastRptRequested	N	N		Removed
Component <Parties>					

ErisX

453	NoPartyIDs	Y	Y	Number of parties.	
448	PartyID	Y	Y	The ID under which the completed trade will be booked. used with on behalf trading, only	
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)	
End Component <Parties>					
17	ExecID	Y	Y	Completed trade identifying number For a chain of partial fills will contain a draft order id	
150	ExecType	Y	Y	The execution report's type. Contains one more value than tag 39, OrderStatus I = Order Status	Removed 0 = New 4 = Canceled 5 = Replaced 8 = Rejected C = Expired F = Trade
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	Removed 2 = Filled 4 = Canceled 8 = Rejected I = Order Status
~~103~~	~~OrdRejReason~~	~~N~~	~~N~~		Removed
55	Symbol	Y	Y	Instrument (E.g. BTC/USD)	
54	Side	Y	Y	1 = Buy 2 = Sell	Removed 7 = Undisclosed
38	OrderQty	Y	Y	The order amount in the currency specified in tag 15	Removed Required for most cases except ExecType = 8
40	OrdType	N	N	2 = Limit order 4 = Stop-Limit order	Removed Required for most cases except ExecType = 8
44	Price	Y	Y	Required for Limit based orders. Price at which the limit order is to be executed.	
99	StopPx	N	Y	Required for Stop orders, OrderType = 4. The	

ErisX

		A		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 this fill. Present when ExecType (150) = F.	
31	LastPx	N	N	Price at which the current or last fill was made. Present when ExecType(150) = F. Not sent for status requests.	
151	LeavesQty	Y	Y	LeavesQty (151) = OrderQty (38) – CumQty (14).	Change. Only working orders are sent.
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 Not sent if ExecType (150) = 8, Rejected, or if the entire order is canceled, ExecType (150) = 4, Canceled.	
75	TradeDate	N	N	Trade date Trades completed after 4 pm CT show the next business day as the trade date.	
59	TimeinForce	Y	Y	How long an order remains in effect: 0=Day 1=Good Till Cancel 3=Immediate of Cancel 4=Fill or Kill 6=Good Till Date (the Expiry Date (432)	
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	Nanosecond Precision
Component <CommissionData>					
12	Commission	N	N	Actual Commission (Only for Fills and Partial Fills)	
7012	CommCalculated	Y	Y	Calculated Commission	
13	CommType	Y	Y	3 = Absolute (Total monetary amount)	
479	CommCurrency	Y	Y	Currency Commission (USD, BTC)	
End Component <CommissionData>					
~~110~~	~~MinQty~~	~~N~~	~~N~~		Removed
~~58~~	~~Text~~	~~N~~	~~N~~		Removed

~~7530~~	~~BelowMin~~	~~N~~	~~N~~		Removed
~~7534~~	~~StopSide~~	~~NA~~	~~N~~		Removed
Standard Trailer					

ErisX

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)
- Complete date: YYYYMMDD (20030716)
- Complete date plus hours and minutes: YYYYMMDD-hh:mm (20030716-19:20)
- Complete date plus hours, minutes and seconds: YYYYMMDD-hh:mm:ss (20030716-19:20:30)
- Complete date plus hours, minutes, seconds and milliseconds YYYYMMDD-hh:mm:ss.mmm (20030716-19:20:30.183)
- Complete date plus hours, minutes, seconds: and microseconds YYYYMMDD-HH:MM:ss.SSSSSS (20180716- 23:15:56.612339)
- Complete date plus 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)