



# **FIX 4.4 STP Specification V3.0**

## **Straight Through Processing (STP)**



Please contact ErisX sales representatives or [technical.onboarding@erisx.com](mailto:technical.onboarding@erisx.com) for more information on this documentation.

## Contents

<b>Change History</b>	<b>3</b>
<b>Introduction</b>	<b>3</b>
Purpose	3
<b>Supported Messages</b>	<b>3</b>
FIX Session Sequence Number Management	3
Session Messages	4
Timestamping / TransactTime (tag 60)	4
Application Messages	4
<b>Message Workflow</b>	<b>5</b>
General Workflow	5
Passwords for Logon Messages	5
<b>Trade Capture Request</b>	<b>6</b>
Trade Capture Report Model	6
Trade Capture Report Request Acknowledgement	6
<b>Trade Capture Reports</b>	<b>6</b>
Trade Report Process Order	6
Duplicate Trade Reports	7
<b>Message Details</b>	<b>7</b>
Session Messages	7
Application Messages	7
TradeCaptureReportRequest	7
TradeCaptureReportRequestAck	8
TradeCaptureReport	8
TradeCaptureReportAck	10



## 1 Change History

Date	Message(s)	Description
20190923	V 2.0	
20191209	V 3.0	Updated to Include details for Futures

## 2 Introduction

### 2.1 Purpose

The purpose of this document is to present in detail the FIX protocol messages for the Straight Through Processing (STP) service which provides guaranteed trade capture reports for an account and can be used for middle and back office reconciliation.

## 3 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.

Members 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.

### 3.1 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 client applications can use the logon confirmation message sequence number to detect a gap since the last received message. If the client application 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. Client applications should avoid submitting subsequent ResendRequest messages. This will simply replace the prior ResendRequest resulting in a delay of normal processing.



## 3.2 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 application, 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.

## 3.3 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.SSSSSSSS.

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

## 3.4 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:

- **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.



- **Trade Capture Report Request** – (In) message used in making a subscription request for trade capture reports. Only subscription requests are currently supported. Individual report request are not supported.
- **Trade Capture Report Request Ack** – (Out) message used to acknowledge receipt of a subscription request. If the request was successful, ErisX will start publishing reports. This message can also be used to reject a report request.
- **Trade Capture Report** – (Out) message used to send completed trade details. Requires confirmation of receipt.
- **Trade Capture Report Ack** – (In) message used to confirm the receipt of a Trade Capture Report.

## 4 Message Workflow

---

### 4.1 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. Any message received by the ErisX FIX STP engine that is outside the scope of this document will be rejected with a Business Message Reject message.

### 4.2 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 ErisX Client Services team.

An expired password must be reset before any application messages. 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.

Users should refer to the ErisX FIX 4.4 Specification document for full details of the Logon and Password messages.



## 5 Trade Capture Request

---

### 5.1 Trade Capture Report Model

A STP client application subscribes to the ErisX FIX Gateway by sending a Trade Capture Report Request message. The results returned depend on the values specified for fields TradeRequestType and SubscriptionRequestType.

- Receive all outstanding unconfirmed trade events and all new subsequent events:
  - TradeRequestType: 569=0 (All trades)
  - SubscriptionRequestType: 570=1 (Snapshot + Updates)
- Receive only new events messages:
  - TradeRequestType: 569=0 (All trades)
  - SubscriptionRequestType: 570=9 (Updates)

### 5.2 Trade Capture Report Request Acknowledgement

ErisX responds to a client application Trade Capture Request with a Trade Capture Report Request Ack message. The status of the request is indicated in the fields TradeRequestResult (749) and TradeRequestStatus (750).

If no TradeCaptureReportRequestAcknowledgement (35=AQ) message is received from ErisX within fifteen (15) seconds of sending a TradeCaptureReportRequest (35=AD) message, the client should disconnect and re-login.

A Trade Capture Report Request acknowledgement containing TradeRequestResult (749) = 0 and TradeRequestStatus (750) = 0 means a successful request has been made.

## 6 Trade Capture Reports

---

### 6.1 Trade Report Process Order

Trade Capture reports are sent in the order in which trades are executed. A new trade event can be sent before a previously sent trade event is acknowledged.

The number of Trade Reports that can be outstanding is limited by a system default. If this limit is reached, a one (1) to one (1) relationship becomes effective in which one (1) Trade Report will be sent for each report acknowledged until the outstanding number of unacknowledged reports is less than the system limit.

- Any reports for trades done prior to the request are sent first.
- Once all Trade Reports from prior to the request, if any, are acknowledged, ErisX will send reports for new trades completed since the Trade Report Request was made.



ErisX will maintain a record of unacknowledged orders for 72 hours. For execution history before that period users should refer to the ErisX Member Portal or use the Clearing WebAPI.

## 6.2 Duplicate Trade Reports

An acknowledged trade cannot be requested again. However, client applications should be able to handle the possibility of receiving duplicate trade capture reports. If multiple reports are received for a trade, all but the first report should be ignored.

## 7 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.

### 7.1 Session Messages

For a description of ErisX FIX Session messages, please refer to the ErisX FIX 4.4 Specification document.Application Messages.



## 7.2 Application Messages

### 7.2.1 TradeCaptureReportRequest

Tag	Field Name	Rqd	New Spec	What's Changed?
Standard Header		Y	<b>MsgType tag 35=AD</b>	
568	TradeRequestID	Y	Id assigned by the client to the request.	
569	TradeRequestType	Y	0 = All trades.	
263	SubscriptionRequestType	Y	1 = Snapshot + Updates 9 = Updates only	Removed 2 = Unsubscribe
55	Symbol	N	"Supported value: "NA" Individual instrument subscription not supported."	Changed "Rqd = N"
167	SecurityType	N	Indicates type of security.	Added Tag 167
<del>1408</del>	<del>DefaultCstmAppVerID</del>	<del>Y</del>		Removed
Standard Trailer		Y		

### 7.2.2 TradeCaptureReportRequestAck

Tag	Field Name	Rqd	New Spec	What's Changed?
Standard Header		Y	<b>MsgType tag 35=AQ</b>	
568	TradeRequestID	Y	Id assigned by the client to the request.	
569	TradeRequestType	Y	0 = All trades.	
263	SubscriptionRequestType	Y	1 = Snapshot + Updates 9 = Updates only	Removed 2 = Unsubscribe
749	TradeRequestResult	Y	0 = Successful 8 = Unsupported TradeRequestType 9 = Unauthorized for Trade Capture Report Request 99 = Other	
750	TradeRequestStatus	Y	0 = Accepted 2 = Rejected	
55	Symbol	N	Supported value: "NA". This will always be set to NA	Changed "Required = N"
167	SecurityType	N	Indicates type of security.	
Standard Trailer		Y		

### 7.2.3 TradeCaptureReport

Tag	Field Name	Rqd	New Spec	What's Changed?
Standard Header		Y	MsgType 35=AE	





15	Currency	Y	The currency for the amount specified in the LastQty (32) field.	
31	LastPx	Y	Trade Price	Updated
32	LastQty	Y	Trade amount in Currency (15)	
Component <Instrument>		Y		
55	Symbol	Y	Instrument (E.g. BTC/USD)	
65	SymbolSfx	Y		Removed
End <Instrument>				
60	TransactTime	Y	Execution Reports will be sent with nanosecond precision - YYYYMMDD-HH:MM:SS.ssssssss	Nanosecond precision
75	TradeDate	Y	Trade Date in YYYYMMDD format.	
150	ExecType	Y	0 = New Trade: generated upon a trade execution	
167	SecurityType	N	FUT = Future, SPOT = Spot	Added
194	LastSpotRate	N	Trade Price	
207	SecurityExchange	N	ERSX	Added
231	ContractMultiplier	N	Specifies the ratio to convert from futures contracts to underlying amount	Added
461	CFIcode	N	Spot = IFXXXP, Futures = FCXXSX	Added
Component <TrdCapRptSideGrp>		Y		
552	NoSides	Y	1 – only value supported	
54	Side	Y	1 = Buy, 2 = Sell	
11	ClOrdID	Y	Last submitted Client Order ID	Updated
1	Account	Y	The account under which the trade was booked	
Component <CommissionData>		Y		
12	Commission	N	Actual Commission	
7012	<del>CommCalculated</del>	Y	Calculated Commission	Removed
13	CommType	Y	3 = Absolute (Total monetary amount)	
479	CommCurrency	Y	Currency Commission (USD, BTC)	
End <CommissionData>		Y		
120	SettlCurrency	N	Counter currency. In Spot, this is the quoted asset	
568	TradeRequestID	Y	Assigned by the client to the request when the report is in response to a Trade Capture Report Request.	
571	TradeReportID	Y	Unique identifier assigned by ErisX to the Trade Capture report.	
828	TrdType	N	Supported Values: 0 = Regular Trade	Removed 102 = ESP



			1 = Block Trade 2 = EFP 3 = Transfer 11 = EFR	Added: 0, 1, 2, 3, 11
1003	TradeID	Y	ErisX assigned trade ID.	
1056	CalculatedCcyLastQty	Y	Trade amount counter currency tag 120	
1057	AggressorIndicator	N	Present on trades with RootPartyRole = 13: Y = Aggressor, N = Aggressed	
<del>880</del>	<del>TradeMatchID</del>	<del>N</del>		Removed
<del>527</del>	<del>SecondaryExecID</del>	<del>N</del>		Removed
<del>570</del>	<del>PreviouslyReported</del>	<del>Y</del>		Removed
Component <RootParties>		Y		
1116	NoRootPartyIDs	Y	Number of RootPartyID's	
1117	RootPartyID	Y	Identifies each party to the trade.	
<del>1118</del>	<del>RootPartyIDSource</del>	<del>N</del>	D = Proprietary code only supported	Removed
1119	RootPartyRole	Y	Role of the party identified in the RootPartyID: 13 = Originating Customer (Account ID) 29 = Intermediary (ErisX) 21 = Clearing Organization 22 = Exchange 44 = Order Entry Operator ID 75 = Location ID	Removed: 1 = Executing Firm
Component <RootSubParties>		N		
1120	NoRootSubParties	N	Number of NoRootSubParties	
1121	RootPartySubID	N	Additional identifier related to RootPartyID (1117). This will be the FIX ID	
1122	RootPartySubID Type	N	Type of RootPartySubID (1121): 2 = Individual Person 32=Execution Venue	

#### 7.2.4 TradeCaptureReportAck

Tag	Field Name	Rqd	New Spec	What's Changed?
Standard Header		Y	<b>MsgType 35=AR</b>	
571	TradeReportID	Y	The TradeReportID (571) from Trade Capture Report being acknowledged.	
55	Symbol	Y	Supported value: "NA".	
Standard Trailer		Y		