



**FIX Drop for Crypto  
Version 2.0**

This specification is being provided to you strictly for informational purposes solely for the purpose of developing or operating systems that interact with EDX Markets, or EDXM. All proprietary rights and interest in this specification and the information contained herein shall be vested in EDXM and all other rights including, but without limitation, patent, registered design, copyright, trademark, service mark, connected with this publication shall also be vested in EDXM. No part of this specification may be redistributed or reproduced in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without written permission from EDXM. EDXM reserves the right to withdraw, modify, or replace the specification at any time, without notice. No obligation is made by EDXM regarding the level, scope, or timing of EDXM's implementation of the functions or features discussed in this specification.

THE SPECIFICATION IS PROVIDED "AS IS", "WITH ALL FAULTS" AND EDXM MAKES NO WARRANTIES AND DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, OR STATUTORY RELATED TO THE SPECIFICATIONS. EDXM IS NOT LIABLE FOR ANY INCOMPLETENESS OR INACCURACIES IN THE SPECIFICATIONS. EDXM IS NOT LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES RELATING TO THE SPECIFICATIONS OR THEIR USE.

## Table of Contents

<b>1</b>	<b>Overview</b>	<b>7</b>
<b>2</b>	<b>Architecture</b>	<b>8</b>
<b>3</b>	<b>Transport Layer</b>	<b>9</b>
<b>4</b>	<b>Session Layer</b>	<b>10</b>
<b>5</b>	<b>Application Layer</b>	<b>11</b>
5.1	Symbology	11
5.2	Order Quantities	11
<b>6</b>	<b>Message Field Types</b>	<b>12</b>
6.1	ExecInst (FIX tag 18)	12
6.2	ExtendedExecInst (FIX tag 9416)	12
6.3	ExecRestatementReason (FIX tag 378)	12
6.4	LastLiquidityInd (FIX tag 851)	12
6.5	OrderCapacity (FIX tag 528)	13
6.6	CustOrderCapacity (FIX tag 582)	13
6.7	OrdStatus (FIX tag 39)	13
6.8	OrdType (FIX tag 40)	13
6.9	SelfTradePreventionType (Custom FIX tag 21001)	14
6.10	Side (FIX tag 54)	14
6.11	STPGroupID (Custom FIX tag 2362)	14
6.12	TimeInForce (FIX tag 59)	15
6.13	CancelReasonCode (Custom FIX tag 21004)	15
6.14	OrdRejReason (FIX tag 103)	16
6.15	CxlRejReason (FIX tag 102)	18
6.16	CxlRejResponseTo (FIX tag 434)	21
6.17	UserStatus (FIX tag 926)	21
6.18	PartyIDSource (FIX tag 447)	21
6.19	PartyRole (FIX tag 452)	22
<b>7</b>	<b>Headers and Trailers</b>	<b>23</b>
7.1	Standard Header	23
7.2	Standard Trailer	23
<b>8</b>	<b>Messages</b>	<b>24</b>
8.1	Session Messages	24
8.2	Heartbeat (Sent by exchange and client)	24
8.3	Logon	24
8.4	Logout	24
8.5	Reject	25
8.6	ResendRequest	25
8.7	SequenceReset	25
8.8	TestRequest	26
<b>9</b>	<b>Session Management</b>	<b>27</b>
<b>10</b>	<b>Application Messages From Exchange to Client</b>	<b>28</b>
10.1	ExecutionReport	28
10.1.1	Trade	28
10.2	Order Drop Execution Reports	28
10.2.1	New Order Acknowledge	29
10.2.2	Rejected Order	30
10.2.3	Replaced	30
10.2.4	Canceled	31

10.2.5 OrderCancelReject ..... 32  
10.2.6 Restatement ..... 32  
10.3 UserNotification..... 33

< THIS PAGE INTENTIONALLY LEFT BLANK >

# 1 Overview

The FIX Drop for Crypto protocol is the application layer message standard used by members to receive client information related to trades executed on the Cryptographic Token Exchange. FIX Drop for Crypto is a text key=value formatted protocol based on the FIX Trading Community FIX 5.0 SP2 specifications.

## 2 Architecture



### 3 Transport Layer

The FIX Drop for Crypto protocol uses the Transmission Control Protocol (TCP) to provide reliable and ordered delivery of messages between clients and servers.

## 4 Session Layer

The FIX Drop for Crypto protocol uses the [FIXT 1.1 specification](#) session messages to establish and maintain a session.

## 5 Application Layer

FIX Drop for Crypto protocol application layer messages are based on the [Financial Information Exchange \(FIX\) Protocol version 5.0](#) and are key=value text encoded.

### 5.1 Symbology

The FIX Drop for Crypto protocol uses a Token ID to uniquely identify a specific instrument. Upon request via a separate channel, the exchange provides a list of instrument directory messages. Each instrument directory message provides, among other information, a Token Name with the format base/quote, (i.e. BTC/USD) and a Token ID: an eight character String that uniquely and permanently identifies the token on the exchange. The Token ID is used in all order related messages.

### 5.2 Order Quantities

Tokens on the exchange trade in lots called a 'unit'. A pre-configured Unit Multiplier is applied to all order quantities for a given token to calculate quantity of the underlying instrument being referenced. The Unit Multiplier is not applied to the price of the order, only the quantity. All quantities referenced in MEMO messages are specified as a whole numbers of units.

For example: a limit order with a quantity of 350 (units) and a limit price of 100.00, for a token "BTC/USD" that was configured with a Unit Multiplier of -8, would imply 100.00 USD for  $350 \times 10^{-8}$  BTC, or 0.00000350 BTC.

Unit Multipliers are constant, configured on the exchange, and will not change during a single logical trading session. The user is required to re-specify the unit multiplier on the order entry protocols as a safety measure to ensure they are trading the quantities they expect to be trading. Any order entered with a unit multiplier that does not match the configured multiplier on the exchange for a given token will be rejected.

## 6 Message Field Types

### 6.1 ExecInst (FIX tag 18)

The ExecInst field provides instructions for order handling on the exchange.

ExecInst is encoded as a MultipleCharValue type. Encoded as a string with one or more values below, separated by spaces.

Value	Name	Description
'6'	ParticipateDoNotInitiate	Post-Only - order will not attempt to remove liquidity on entry.

### 6.2 ExtendedExecInst (FIX tag 9416)

Designates that an order is a Retail order or a Retail Liquidity Providing order.

ExtendedExecInst is encoded as a MultipleCharValue type. Encoded as a string with one or more values below, separated by spaces.

Value	Name	Description
'R'	DesignatedRetail	Designated Retail Only - allowed to interact with Retail Liquidity Provider Orders.
'T'	RetailLiquidityProvider	Order Providing Liquidity For Designated Retail Orders Only.

### 6.3 ExecRestatementReason (FIX tag 378)

Defines the reason for a restatement.

ExecRestatementReason is encoded as an int type.

Value	Name	Description
1	GTT renewal / restatement	GTT order is reinstated on the book following a maintenance window.
5	SelfTradePrevention	The order quantity was reduced to prevent a wash trade.

### 6.4 LastLiquidityInd (FIX tag 851)

Indicates the liquidity code for this execution.

NOTE: Values 70, 71, 170, and 171 are custom values.

LastLiquidityInd is encoded as an int type.

Value	Name	Description
1	AddDisplayed	The order added displayed liquidity to the Exchange book.
2	Removed	The order removed liquidity from the Exchange book.
70	RetailLiquidityAdded	The order available for retail access only added liquidity to the Exchange book.
71	RetailLiquidityRemoved	The order available for retail access only removed liquidity from the Exchange book.

Value	Name	Description
170	DesignatedRetailAdded	The designated retail only order added liquidity to the Exchange book.
171	DesignatedRetailRemoved	The designated retail only order removed liquidity from the Exchange book.

## 6.5 OrderCapacity (FIX tag 528)

Designates the capacity of the firm placing the order.

OrderCapacity is encoded as a char type.

Value	Name
'A'	Agency
'P'	Principal
'R'	Riskless Principal

## 6.6 CustOrderCapacity (FIX tag 582)

Capacity of customer placing the order.

CustOrderCapacity is encoded as an int type.

Value	Name
5	RetailCustomer
1	MemberTradingOnTheirOwnAccount

## 6.7 OrdStatus (FIX tag 39)

Indicates the status of an order.

OrdStatus is encoded as a char type.

Value	Name
'0'	New
'1'	PartialFilled
'2'	Filled
'4'	Canceled
'6'	PendingCancel
'8'	Rejected
'A'	PendingNew
'E'	PendingReplace
'C'	Expired

## 6.8 OrdType (FIX tag 40)

Indicates the type of an order.

OrdType is encoded as a char type.

Value	Name
'1'	Market
'2'	Limit

## 6.9 SelfTradePreventionType (Custom FIX tag 21001)

Defines the self trade operation which will be used if the exchange self-trade-prevention functionality is triggered on this order.

If this order would execute against opposite side resting interest with the same self-trade prevention identifier, the behavior specified by this value will be triggered.

SelfTradePreventionType is encoded as an int type.

Value	Name	Description
0	CancelNewest	If triggered, this order (the incoming order) will be canceled.
1	CancelOldest	If triggered, the resting order will be canceled.
2	N/A	Reserved value.
3	CancelBoth	If triggered, both the incoming and the resting orders will be canceled.

## 6.10 Side (FIX tag 54)

Defines the side of an order.

Side is encoded as a char type.

Value	Name
'1'	Buy
'2'	Sell

## 6.11 STPGroupID (Custom FIX tag 2362)

This field uniquely identifies the self trade prevention group. Several of the values are reserved values while other values are treated as custom values and are at the discretion of the client.

STPGroupID is encoded as an int type.

Value	Name	Description
0	Firm Scope	Prevents orders from the same firm trading with each other.
1	CPID Scope	Prevents orders from the same Crypto Participant ID trading with each other.
2	Account Scope	Prevents orders from the same SenderCompID trading with each other.
3-65534	Custom Scope	Prevents orders with matching STPGroupID across the firm from trading with each other.

## 6.12 TimeInForce (FIX tag 59)

Defines the time during which an order is eligible for execution.

TimeInForce is encoded as a char type.

Value	Name	Description
'3'	Immediate Or Cancel (IOC)	The order shall be partially or completely executed immediately, or canceled outright.
'4'	Fill Or Kill (FOK)	The order shall be immediately executed in full or canceled outright.
'A'	GoodForTime	The order shall be eligible for execution during the supplied time interval.

## 6.13 CancelReasonCode (Custom FIX tag 21004)

Identifies the reason an order was canceled by the system.

CancelReasonCode is encoded as an int type.

Value	Name	Description
0	Other	This order was canceled for some other reason not listed.
1	UserRequestedCancel	The client sent a OrderCancelRequest for this order.
2	PriceExceedsCurrentPriceBand	The price of the order fell outside execution price collar bands.
3	Halted	The market on the order's security was halted.
4	ExchangeSupervisory	Operational or supervisory actions taken by the Exchange resulted in the cancellation of this order.
5	OrderExpired	The order was sent with an expiration time and had the "good for time" time in force set, and the supplied expiration time passed.
6	ParticipantDisconnect	The participant directed that their orders should be canceled when the trading system detects a disconnection, and the participant disconnected.
7	OrderNotBookable	Order is not of bookable type (this may include market orders, IOC, etc).
8	FirmDisabled	Order was canceled because the firm was disabled.
9	CPIDDisabled	Order was canceled because the Crypto Participant ID was disabled.
10	AccountDisabled	Order was canceled because the Account for the order port was disabled.
11	NotionalExposureRiskBreach	Order was canceled because a Notional Exposure Risk Rule was breached.
12	InsufficientCredit	Order was canceled due to lack of credit with a counterparty during an execution action.
13	OrderLocksOrCrosses	Order was canceled because it would lock or cross the displayed order book.
14	OrderCannotBeFullyFilled	A Fill-Or-Kill or Immediate-Or-Cancel order cannot execute for its full quantity upon entry and is being canceled.

## 6.14 OrdRejReason (FIX tag 103)

Identifies the reason why a new order was rejected by the system.

Values 0-99 are reserved by the FIX specification. Note that not all of the reserved FIX codes are used.

Values 100-199 are used to denote issues with specific request fields.

Values 200-299 are used to denote issues with the request as a whole.

Values 300-399 are used to denote issues related to risk.

Also note that some codes in this table may not be sent in FIX but are listed here to maintain consistency with the SBE order entry specification, or reserved for future use.

OrdRejReason is encoded as an int type.

Value	Name	Description
1	UnknownSymbol	The order contained a valid token identifier, but the token is not currently traded on this port or exchange.
2	ExchangeClosed	The exchange is closed and not accepting orders at this time.
6	DuplicateOrder	The user entered a duplicate order identifier.
16	PriceExceedsCurrentPriceBand	The price of the order fell outside execution price collar bands.
18	InvalidPriceIncrement	The order contained an invalid limit price increment.
99	Other	This order was rejected for some other reason not listed.
100	MissingTokenID	The order did not supply the TokenID.
101	InvalidTokenID	A TokenID was present but contained illegal characters. NOTE: this code is only used when the field is malformed. If the TokenID is otherwise well-formed, but the Token it refers to is not traded on the port or exchange, the standard FIX code UnknownSymbol (1) will be sent.
102	MissingCIOrdID	The order did not supply the CIOrdID.
103	InvalidCIOrdID	A CIOrdID was present but contained illegal characters.
104	MissingSide	The order did not supply the side.
105	InvalidSide	The order contained an invalid side.
106	MissingOrderQty	The order did not supply the quantity.
107	InvalidOrderQty	The order contained an invalid order quantity.
108	MissingOrderType	The order did not supply the order type.
109	InvalidOrderType	The order contained an invalid order type.
110	MissingTimeInForce	The order did not supply the time in force.
111	InvalidTimeInForce	The order contained an invalid time in force.
112	MissingOrderCapacity	The order did not supply the order capacity.
113	InvalidOrderCapacity	The order contained an invalid order capacity.
114	MissingExecInst	The order did not supply the execution instructions field.
115	InvalidExecInst	The order contained an invalid execution instruction.
116	MissingExtendedExecInst	The order did not supply the extended execution instructions field.
117	InvalidExtendedExecInst	The order contained an invalid extended execution instruction.



Value	Name	Description
118	MissingLimitPrice	The order did not supply the limit price.
119	InvalidLimitPrice	The order contained an invalid limit price.
120	MissingCustomerCapacity	The order did not supply the customer capacity.
121	InvalidCustomerCapacity	The order contained an invalid customer capacity.
122	MissingExpireTime	The order did not supply the expire time.
123	InvalidExpireTime	The order contained an invalid expire time.
124	MissingSelfTradePreventionType	The order supplied the STPGroupID field but did not supply the SelfTradePreventionType field.
125	InvalidSelfTradePreventionType	The order contained an invalid SelfTradePreventionType.
126	MissingCancelGroupID	The order did not supply the CancelGroupID field (used when a port is configured with a fixed set of allowable CancelGroupIDs).
127	InvalidCancelGroupID	The order contained an invalid CancelGroupID (used when a port is configured with a fixed set of allowable CancelGroupIDs).
128	MissingSTPGroupID	The order did not supply the STPGroupID field (used when a port is configured with a fixed set of allowable STPGroupIDs).
129	InvalidSTPGroupID	The order contained an invalid STPGroupID (used when a port is configured with a fixed set of allowable STPGroupIDs).
130	MissingLnkID	The order did not supply the LnkID field (currently unused - reserved for consistency / future usage).
131	InvalidLnkID	The order was rejected for an invalid LnkID (value contained non-printable ASCII characters).
132	MissingUnitMultiplier	The order did not supply the unit multiplier.
133	InvalidUnitMultiplier	The supplied unit multiplier did not match the configured multiplier on the exchange.
134	MissingRiskGroupID	The order did not supply the RiskGroupID field (used when a port is configured with a fixed set of allowable RiskGroupIDs).
135	InvalidRiskGroupID	The order contained an invalid RiskGroupID (used when a port is configured with a fixed set of allowable RiskGroupIDs).
136	MissingCPID	The order did not supply the CPID field.
137	InvalidCPID	The order contained an invalid CPID.
138-199	N/A	Reserved.
200	Halted	The token supplied in the order is currently halted.
201	FirmDisabled	The order was rejected because the firm sending the order has been disabled.
202	CPIDDisabled	The order was rejected because the Crypto Participant ID used in the order has been disabled.
203	AccountDisabled	The order was rejected because the account sending the order has been disabled.
204	InvalidModifierForOrderType	The order contained one or more modifiers that are not compatible with the order type provided.
205	InvalidTimeInForceForOrderType	The order contained a time in force that is not compatible with the order type provided.
206	InvalidOrderTypeForTradingState	The order type is not accepted during the current trading state for the token. (Market order sent during Quoting or LimitOnlyTrading states).

Value	Name	Description
207	PostOnlyNotAllowed	Post-only behavior is not allowed with one or more of the supplied order directives.
208-299	N/A	Reserved.
300	NonTestTokensBlocked	The order was for a non-test token and the port is configured in test-only mode (exchange configuration).
301	NotionalValueExceedsThreshold	The notional value of the order exceeded an exchange configured maximum.
302	MaxNotionalValuePerOrder RiskRuleViolated	The notional value of the order violated the firm's max notional value per order risk rule.
303	BlockNonTestToken RiskRuleViolated	The order violated the firm's test token trading risk rule.
304	MaxSizePerOrder RiskRuleViolated	The order breached the firm's max tokens per order risk rule.
305	PricePercentCollar RiskRuleViolated	The order violated the firm's price percent collar risk rule.
306	PriceValueCollar RiskRuleViolated	The order violated the firm's price value collar risk rule.
307	MaxADVPercentPerOrder RiskRuleViolated	The order violated the firm's max ADV (average daily volume) percentage risk rule.
308	MaxNumDuplicateOrders RiskRuleBreach	The system detected a breach in the firm's max number of duplicate orders risk rule. Orders are being rejected until the breach is manually cleared.
309	MaxOrderRate RiskRuleBreach	The system detected a breach in the firm's max number of orders per second risk rule. Orders are being rejected until the breach is manually cleared.
310	GrossNotionalExposure RiskRuleBreach	The system detected a breach in the firm's gross notional exposure risk rule. Orders are being rejected until the breach is manually cleared.
311	NetNotionalExposure RiskRuleBreach	The system detected a breach in the firm's net notional exposure risk rule. Orders are being rejected until the breach is manually cleared.
312	MarketOrder GrossNotionalExposure RiskRuleBreach	The system detected a breach in the firm's market order gross notional exposure risk rule. Orders are being rejected until the breach is manually cleared.
313	MarketOrder NetNotionalExposure RiskRuleBreach	The system detected a breach in the firm's market order net notional exposure risk rule. Orders are being rejected until the breach is manually cleared.

## 6.15CxlRejReason (FIX tag 102)

Identifies the reason why an order cancellation or replacement request was rejected.

Values 0-99 are reserved by the FIX specification. Note that not all of the reserved FIX codes are used.

Values 100-199 are used to denote issues with specific request fields.

Values 200-299 are used to denote issues with the request as a whole.

Values 300-399 are used to denote issues related to risk.

Also note that some codes in this table may not be sent in FIX but are listed here to maintain consistency with the SBE order entry specification, or reserved for future use.

CxlRejReason is encoded as an int type.

Value	Name	Description
0	TooLateToCancel	A cancel or replace request was received after the exchange has closed.
1	UnknownOrder	An order with the specified identifier was not found.
3	OrderAlreadyPendingCancelOrReplace	There is a pending request on this order.
6	DuplicateClOrdID	The supplied ClOrderID duplicated another order for this account (used for replace requests).
8	PriceExceedsCurrentPriceBand	The price of the replacement order fell outside execution price collar bands.
18	InvalidPriceIncrement	Request Limit Price increment is invalid.
99	Other	This request was rejected for some other reason not listed.
100	MissingTokenID	The request did not supply a Token ID.
101	InvalidTokenID	The request supplied an invalid Token ID.
102	MissingClOrdID	The request did not supply a ClOrdID.
103	InvalidClOrdID	The request supplied an invalid ClOrdID.
104	MissingSide	The request did not supply a side.
105	InvalidSide	The request supplied an invalid side.
106	MissingOrderQty	The request did not supply a quantity.
107	InvalidOrderQty	The request supplied a quantity, but it was set to an invalid value (i.e. 0, larger than exchange max, etc).
108	MissingOrderType	The request did not supply an order type.
109	InvalidOrderType	The request supplied an invalid order type.
110	MissingLimitPrice	The request did not supply a LimitPrice field and was required to due to OrderType requested.
111	InvalidLimitPrice	The request supplied the LimitPrice field but the value was invalid (0, more than the max allowed by the exchange, etc) or not applicable (i.e. the OrderType was set to Market).
112	MissingLnkID	The request did not supply the LnkID field. (currently unused as this field is optional - reserved for future use if needed to keep specification values consistent)
113	InvalidLnkID	The request supplied the LnkID field, but it was too large, contained illegal characters, etc.
114	MissingQuoteIndex	The request did not supply the QuoteIndex field.
115	InvalidQuoteIndex	The request supplied the QuoteIndex field, but it was set to an invalid value.
116	MissingOrigClOrdID	The request did not supply the OrigClOrdID field.
117	InvalidOrigClOrdID	The request supplied the OrigClOrdID field, but it was too large, contained illegal characters, etc.
118	MissingOrderID	The request did not supply the OrderID field.

Value	Name	Description
119	InvalidOrderID	The request supplied the OrderID field, but it was too large, contained illegal characters, etc.
120-199	N/A	Reserved.
200	Halted	The token supplied in the request is currently halted.
201	FirmDisabled	The request was rejected because the firm sending the order has been disabled.
202	CPIDDisabled	The request was rejected because the Crypto Participant ID used in the order has been disabled.
203	AccountDisabled	The request was rejected because the account sending the order has been disabled.
204	UnsupportedOrdTypeChange	Change of OrdType in replace request not supported.
205	UnsupportedSideChange	Change of side in replace request not supported.
206	TokenMismatch	Token provided on a replace request does not match the token on the original order associated with the provided OrigClOrdId or OrderID.
207	OrigOrderIDMismatch	OrigClOrdID and OrderID were both provided, but the provided OrderID does not match the actual OrderID associated with the provided OrigClOrdID.
208-299	N/A	Reserved.
300	NonTestTokensBlocked	The request was for a non-test token and the port is configured in test-only mode (exchange configuration).
301	NotionalValueExceedsThreshold	The notional value of the replace request exceeded an exchange configured maximum.
302	MaxNotionalValuePerOrder RiskRuleViolated	The notional value of the replace request violated the firm's max notional value per order risk rule.
303	BlockNonTestToken RiskRuleViolated	The replace request violated the firm's test token trading risk rule.
304	MaxSizePerOrder RiskRuleViolated	The replace request breached the firm's max tokens per order risk rule.
305	PricePercentCollar RiskRuleViolated	The replace request violated the firm's price percent collar risk rule.
306	PriceValueCollar RiskRuleViolated	The replace request violated the firm's price value collar risk rule.
307	MaxADVPercentPerOrder RiskRuleViolated	The replace request violated the firm's max ADV (average daily volume) percentage risk rule.
308	MaxNumDuplicateOrders RiskRuleBreach	The system detected a breach in the firm's max number of duplicate orders risk rule. Orders are being rejected until the breach is manually cleared.
309	MaxOrderRate RiskRuleBreach	The system detected a breach in the firm's max number of orders per second risk rule.

Value	Name	Description
		Orders are being rejected until the breach is manually cleared.
310	GrossNotionalExposure RiskRuleBreach	The system detected a breach in the firm's gross notional exposure risk rule. Orders are being rejected until the breach is manually cleared.
311	NetNotionalExposure RiskRuleBreach	The system detected a breach in the firm's net notional exposure risk rule. Orders are being rejected until the breach is manually cleared.
312	MarketOrder GrossNotionalExposure RiskRuleBreach	The system detected a breach in the firm's market order gross notional exposure risk rule. Orders are being rejected until the breach is manually cleared.
313	MarketOrder NetNotionalExposure RiskRuleBreach	The system detected a breach in the firm's market order net notional exposure risk rule. Orders are being rejected until the breach is manually cleared.

## 6.16 CxlRejResponseTo (FIX tag 434)

Identifies the request type that a cancel reject was issued for.

CxlRejResponseTo is encoded as a char type.

Value	Name	Description
'1'	OrderCancelRequest	This response is for a cancel request.
'2'	OrderCancelReplaceRequest	This response is for a replace request.

## 6.17 UserStatus (FIX tag 926)

Defines the changed with the session state as it pertains the User.

Values 0-99 are reserved by the FIX specification. Note that not all of the reserved FIX codes are used.

UserStatus is encoded as an int type.

Value	Name
8	Session shutdown warning.
100	Session shutdown, end of events for session.

## 6.18 PartyIDSource (FIX tag 447)

Identifies the class or source of the PartyID (Tag 448) Value.

PartyIDSource is encoded as a char type.

Value	Name
'C'	Generally accepted market participant identifier.

## 6.19 PartyRole (FIX tag 452)

Identifies the type or role of the PartyID (Tag 448) Value.

PartyRole is encoded as an int type.

Value	Name
12	Executing Trader

## 7 Headers and Trailers

All messages begin with a *Standard Header* and must end with a *Standard Trailer*.

### 7.1 Standard Header

The *Standard Header* must contain the required fields listed and can contain any of the non-required listed fields:

Tag	Field Name	Req'd	Meaning/Values
8	BeginString	Y	"FIXT.1.1" This must be the first field in the message.
9	BodyLength	Y	The length of the FIX message. This must be the second field in the message.
35	MsgType	Y	The type of message. This must be the third field in the message.
49	SenderCompID	Y	The sender's id, assigned by at on-boarding time.
56	TargetCompID	Y	"EDXM"
34	MsgSeqNum	Y	The message sequence number.
52	SendingTime	Y	The time of message transmission in Universal Time coordinated. See UTCTimestamp in FIX 5.0 SP2 specification.
43	PossDupFlag	N	Always required for retransmitted messages, whether prompted by the sending system or as the result of a resend request.
97	PossResend	N	Required when message may be duplicate of another message sent under a different sequence number.
122	OrigSendingTime	N	Required for message resent as a result of a ResendRequest. If data is not available set to same value as SendingTime.

### 7.2 Standard Trailer

The *Standard Trailer* must contain the CheckSum field.

Tag	Field Name	Req'd	Meaning/Values
10	Checksum	Y	The message checksum. This must be the last field in the message.

## 8 Messages

This section defines the messages that make up the protocol. For each message, it lists the fields in the message by tag id and name, whether the field is required and acceptable values or description of the field.

### 8.1 Session Messages

This section defines the session level messages supported.

### 8.2 Heartbeat (Sent by exchange and client)

The Heartbeat message confirms the status of the communication link.

Tag	Field Name	Req'd	Meaning/Values
	<a href="#">Standard Header</a>	Y	With tag 35 (MsgType) = 0
112	TestReqID	N	Identifier included in Test Request message to be returned in resulting Heartbeat. (Max Length: 64 characters)
	<a href="#">Standard Trailer</a>	Y	

### 8.3 Logon

The logon message is sent by the client to establish a connection to the exchange.

A logon message must be the first message sent by the application initiating a FIX session.

Tag	Field Name	Req'd	Meaning/Values
	<a href="#">Standard Header</a>	Y	With tag 35 (MsgType) = A
98	EncryptMethod	Y	Required to be 0.
108	HeartBtInt	Y	Heartbeat interval, in seconds. An interval of 0 indicates there are no heartbeats. The maximum supported interval is 90 seconds.
141	ResetSeqNumFlag	N	FIX Drop Copy will reject logon if this flag is 'Y'. Market Operations can reset the inbound and outbound sequence numbers on request.
1137	DefaultAppVerID	Y	Required to be 9. ("FIX50SP2")
1408	DefaultCstmAppVerID	Y	The schema version is used to ensure consistency across message formats. Required to be set to "2.0"
	<a href="#">Standard Trailer</a>	Y	

Note: If the DefaultCstmAppVerID (tag 1408) is invalid, the client connection is dropped.

### 8.4 Logout

The logout message initiates or confirms the termination of a FIX session.

This message may be sent by the exchange or by the client.



Tag	Field Name	Req'd	Meaning/Values
	<a href="#">Standard Header</a>	Y	With tag 35 (MsgType) = 5
58	Text	N	Text explaining reason for logout. (Max Length: 128 characters)
	<a href="#">Standard Trailer</a>	Y	

## 8.5 Reject

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

Tag	Field Name	Req'd	Meaning/Values
	<a href="#">Standard Header</a>	Y	With tag 35 (MsgType) = 3
45	RefSeqNum	Y	Reference message sequence number.
371	RefTagID	N	The tag number of the FIX field that was rejected.
372	RefMsgType	N	The MsgType of the FIX message that was rejected.
58	Text	N	Explanatory reason. (Max Length: 128 characters)
	<a href="#">Standard Trailer</a>	Y	

## 8.6 ResendRequest

The resend request is sent by the receiving application to initiate the retransmission of messages.

Tag	Field Name	Req'd	Meaning/Values
	<a href="#">Standard Header</a>	Y	With tag 35 (MsgType) = 2
7	BeginSeqNo	Y	Message sequence number of first message in range to be resent.
16	EndSeqNo	Y	Message sequence number of last message in range to be resent.
	<a href="#">Standard Trailer</a>	Y	

## 8.7 SequenceReset

The sequence reset message is used to reset the incoming sequence number on the opposing side.

Tag	Field Name	Req'd	Meaning/Values
	<a href="#">Standard Header</a>	Y	With tag 35 (MsgType) = 4
123	GapFillFlag	N	Indicates that this message is replacing administrative or application messages which will not be resent.
36	NewSeqNo	Y	New sequence number.
	<a href="#">Standard Trailer</a>	Y	

## 8.8 TestRequest

The test request message forces a heartbeat from the opposing application.

The opposing application responds to the Test Request with a Heartbeat containing the TestReqID.

Tag	Field Name	Req'd	Meaning/Values
	<a href="#">Standard Header</a>	Y	With tag 35 (MsgType) = 1
112	TestReqID	Y	Identifier included in Test Request message to be returned in resulting Heartbeat. (Max Length: 64 characters)
	<a href="#">Standard Trailer</a>	Y	

## 9 Session Management

The client establishes a session by opening a TCP connection to the exchange FIX Drop port (server) and sending a FIX logon message with the appropriate credentials and MsgSeqNum. The SenderCompID and TargetCompID will be provisioned by the exchange and will be verified by the server at Logon. If the Logon message is accepted, the server will send a logon response. If the MsgSeqNum on the inbound logon is less than expected by the server, the logon will be rejected.

If the MsgSeqNum of the inbound Logon message is greater than expected, the server will reply with a Logon, and shall send a ResendRequest for the missing messages. The client application should gap fill any session level messages (see [FIXT 1.1 specification](#) Message Recovery), but may opt to issue sequence reset messages with the GapFillFlag set.

There are no client to server business level messages specified in this Drop Copy API specification. When the client application receives the Logon response, it should compare the MsgSeqNum to the expected server sequence number. If a gap is detected, it should send a ResendRequest message to the server. The server will resend any missing business level messages with the PossDup flag set. Instead of sending session-level messages during a backfill, the server will issue sequence reset messages with the GapFillFlag set.

When the client terminates a session it should send a Logout message and wait for the Logout response from the server before closing the connection. This procedure facilitates gap detection and processing before the connection is closed.

At the end of a logical trading session, the FIX inbound and outbound sequence numbers are set to 1. All inbound FIX messages will be rejected if a FIX session has not been established. If the server receives a message with a MsgSeqNum less than the expected client sequence number, and that message does not have the PossDup flag set, the server will disconnect the client. In the event of a failover to a backup port, the backup port will maintain the sequence numbers of the primary, and it will have the ability to replay any messages that the primary port sent.

## 10 Application Messages From Exchange to Client

### 10.1 ExecutionReport

#### 10.1.1 Trade

This execution report is sent when a trade occurs.

Tag	Field Name	Req'd	Meaning/Values
	<a href="#">Standard Header</a>	Y	With tag 35 (MsgType) = 8
37	OrderID	Y	Unique identifier for an Order assigned by the exchange.
11	ClOrdID	Y	Unique identifier for an Order as assigned by the client.
17	ExecID	Y	Unique identifier of execution message as assigned by the exchange.
150	ExecType	Y	F - Trade (partial fill or fill).
39	OrdStatus	Y	1 - Partially filled 2 - Filled
55	Symbol	Y	As stated in the order.
54	Side	Y	As stated in the order.
21023	QuoteIndex	N	As stated in NewOrderBulk request or 0 for NewOrderSingle request.
32	LastQty	Y	Quantity bought/sold on this (last) fill. See <a href="#">Order Quantities</a> for more details.
31	LastPx	Y	Price of this (last) fill.
151	LeavesQty	Y	Quantity open for execution. See <a href="#">Order Quantities</a> for more details.
14	CumQty	Y	Total quantity filled on the order. See <a href="#">Order Quantities</a> for more details.
44	Price	N	As stated in the order.
38	OrderQty	Y	As stated in the order. See <a href="#">Order Quantities</a> for more details.
40	OrdType	Y	As stated on the previous order.
60	TransactTime	Y	The time at which the transaction occurred.
880	TrdMatchID	Y	Identifier assigned to the Trade by the matching system.
851	LastLiquidityInd	Y	Indicator denoting whether the referenced order removed liquidity from or added liquidity to the Exchange book.
1	Account	Y	Account associated to the order.
453	NoPartyIDs	Y	Number of PartyID's to follow (Always 1).
447	PartyIDSource	Y	Identifies class or source of the PartyID (448) value.
448	PartyID	Y	CPID as stated in the order, or the default CPID for the Account if not specified in the order.
452	PartyRole	Y	Identifies the type or role of the PartyID (448) specified.
583	LnkID	N	As stated in the order.
	<a href="#">Standard Trailer</a>	Y	

### 10.2 Order Drop Execution Reports

FIX Drop Copy by default will always send Trade execution reports. If clients opt in for Order Drop, the port will also disseminate the order entry execution reports listed below.

## 10.2.1 New Order Acknowledge

The New Order Acknowledge execution report is a response to the NewOrderSingle message in the event the order was accepted by the exchange. This message echos back to the requester the accepted state of the order.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 8
37	OrderID	Y	Unique identifier for an Order assigned by the exchange.
11	ClOrdID	Y	As stated in the order.
17	ExecID	Y	Unique identifier of execution message as assigned by the exchange.
21025	CorrelationID	Y	A reference number used to match this order with orders broadcast on MEMOIR Depth for Crypto. Values will be unique per token, per trading session.
150	ExecType	Y	0 - New
39	OrdStatus	Y	0 - New
55	Symbol	Y	As stated in the order.
21023	QuoteIndex	N	As stated in NewOrderBulk request or 0 for NewOrderSingle request.
21024	UnitMultiplier	Y	As stated in the order.
54	Side	Y	As stated in the order.
151	LeavesQty	Y	Quantity open for further execution. See Order Quantities for more details.
14	CumQty	Y	0
44	Price	N	As stated in the order.
38	OrderQty	Y	As stated in the order. See Order Quantities for more details.
40	OrdType	Y	As stated in the order.
59	TimeInForce	Y	As stated in the order.
528	OrderCapacity	Y	As stated in the order.
18	ExecInst	Y	As stated in the order.
9416	ExtendedExecInst	N	As stated in the order.
126	ExpireTime	N	As stated in the order.
21001	SelfTradePreventionType	N	As stated in the order. This is a custom FIX tag.
2362	STPGroupID	N	As stated in the order. This is a custom FIX tag.
21005	RiskGroupID	N	As stated in the order. This is a custom FIX tag.
582	CustOrderCapacity	Y	As Stated in the order.
60	TransactTime	Y	The time at which the transaction occurred. UTC timestamp since epoch with millisecond precision.
1	Account	Y	Account associated to the order.
453	NoPartyIDs	Y	Number of PartyID's to follow (Always 1).
447	PartyIDSource	Y	Identifies class or source of the PartyID (448) value.
448	PartyID	Y	CPID as stated in the order, or the default CPID for the Account if not specified in the order.
452	PartyRole	Y	Identifies the type or role of the PartyID (448) specified.
583	LnkID	N	As stated in the order.
	Standard Trailer	Y	

## 10.2.2 Rejected Order

The Rejected Order execution report is sent in response to a NewOrderSingle message that was rejected by the exchange.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 8
11	ClOrdID	Y	As stated in the order.
37	OrderID	Y	Unique identifier for an Order assigned by the exchange, if available.
17	ExecID	Y	Unique identifier of execution message as assigned by the exchange.
150	ExecType	Y	8 - Rejected
39	OrdStatus	Y	8 - Rejected
55	Symbol	Y	As stated in the order.
54	Side	Y	As stated in the order.
21023	QuoteIndex	N	As stated in NewOrderBulk request or 0 for NewOrderSingle request.
151	LeavesQty	Y	0
14	CumQty	Y	0
44	Price	N	As stated in the order.
38	OrderQty	Y	As stated in the order. See Order Quantities for more details.
40	OrdType	Y	As stated in the order.
60	TransactTime	Y	UTC timestamp since epoch with millisecond precision.
103	OrdRejReason	Y	Reason code for order rejection.
1	Account	Y	Account associated to the order.
453	NoPartyIDs	Y	Number of PartyID's to follow (Always 1).
447	PartyIDSource	Y	Identifies class or source of the PartyID (448) value.
448	PartyID	Y	CPID as stated in the order, or the default CPID for the Account if not specified in the order.
452	PartyRole	Y	Identifies the type or role of the PartyID (448) specified.
583	LnkID	N	As stated in the order.
	Standard Trailer	Y	

## 10.2.3 Replaced

The Replaced execution report is sent when an order has been replaced.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 8
37	OrderID	Y	Unique identifier for an Order assigned by the exchange.
11	ClOrdID	Y	As stated in the replace request. Will be applied as the new ClOrdID for this order going forward.
41	OrigClOrdID	Y	The ClOrdID associated with the replaced order.
17	ExecID	Y	Unique identifier of execution message as assigned by the exchange.
21025	CorrelationID	Y	A reference number used to match this order with orders broadcast on MEMOIR Depth for Crypto. Values will be unique per token, per trading session.
150	ExecType	Y	5 - Replaced
39	OrdStatus	Y	0 - New 1 - PartiallyFilled

Tag	Field Name	Req'd	Meaning/Values
55	Symbol	Y	As stated in the replace request.
54	Side	Y	As stated in the replace request.
21023	QuoteIndex	N	As stated in NewOrderBulk request or 0 for NewOrderSingle request.
151	LeavesQty	Y	Quantity open for execution. See Order Quantities for more details.
14	CumQty	Y	Total quantity filled on the order. See Order Quantities for more details.
44	Price	N	As stated in the replace request.
38	OrderQty	Y	As stated in the replace request. See Order Quantities for more details.
40	OrdType	Y	As stated in the replace request.
59	TimeInForce	Y	As stated in the replace request.
60	TransactTime	Y	UTC timestamp since epoch with millisecond precision.
1	Account	Y	Account associated to the order.
453	NoPartyIDs	Y	Number of PartyID's to follow (Always 1).
447	PartyIDSource	Y	Identifies class or source of the PartyID (448) value.
448	PartyID	Y	CPID as stated in the order, or the default CPID for the Account if not specified in the order.
452	PartyRole	Y	Identifies the type or role of the PartyID (448) specified.
583	LnkID	N	As stated on the replace request, or the original order if not supplied on the replace request.
	Standard Trailer	Y	

### 10.2.4 Canceled

The Canceled execution report is sent when an order has been canceled. This may be because of a client requested cancel, a replace, or a cancel initiated by the exchange. Orders that are canceled by the exchange because of Time in Force instructions shall have an ExecType and OrdStatus as Expired. All other cases shall have ExecType and OrdStatus as Canceled.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 8
37	OrderID	Y	Unique identifier for an Order assigned by the exchange.
11	ClOrdID	Y	As stated in the cancel request or the ClOrdID of the order if an unsolicited cancel.
41	OrigClOrdID	Y	The ClOrdID associated with the canceled order.
17	ExecID	Y	Unique identifier of execution message as assigned by the exchange.
150	ExecType	Y	4 - Canceled C - Expired
39	OrdStatus	Y	4 - Canceled C - Expired
55	Symbol	Y	As stated in the cancel request.
54	Side	Y	As stated in the cancel request.
21023	QuoteIndex	N	As stated in NewOrderBulk request or 0 for NewOrderSingle request.
151	LeavesQty	Y	0
14	CumQty	Y	Total quantity filled on the order. See Order Quantities for more details.

Tag	Field Name	Req'd	Meaning/Values
44	Price	N	As stated on the order.
38	OrderQty	Y	As stated on the order. See Order Quantities for more details.
21004	CancelReason	Y	Reason code for order cancellation. This is a custom FIX tag.
60	TransactTime	Y	UTC timestamp since epoch with millisecond precision.
1	Account	Y	Account associated to the order.
453	NoPartyIDs	Y	Number of PartyID's to follow (Always 1).
447	PartyIDSource	Y	Identifies class or source of the PartyID (448) value.
448	PartyID	Y	CPID as stated in the order, or the default CPID for the Account if not specified in the order.
452	PartyRole	Y	Identifies the type or role of the PartyID (448) specified.
583	LnkID	N	As stated in the order.
	Standard Trailer	Y	

### 10.2.5 OrderCancelReject

The order cancel reject message is used by the exchange to indicate a cancel or a cancel/replace request cannot be honored.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 9
37	OrderID	Y	Unique identifier for an Order as assigned by the exchange. If the ClOrdID on the CancelRequest is unknown this field will contain "NONE".
11	ClOrdID	Y	Unique identifier of the cancel or cancel/replace request being rejected. This field is restricted to the use of uppercase alphanumerics.
54	Side	Y	As stated in the cancel request.
21023	QuoteIndex	N	As stated in NewOrderBulk request or 0 for NewOrderSingle request.
41	OrigClOrdID	Y	As specified on cancel/replace request.
39	OrdStatus	Y	The status of the order after this reject was applied.
434	CxlRejResponseTo	Y	Identifies the type of request that a Cancel Reject is in response to.
102	CxlRejReason	Y	Code to identify reason for cancel rejection.
583	LnkID	N	As stated in the order as long as the order is a live order.
	Standard Trailer	Y	

### 10.2.6 Restatement

The Restatement execution report is an unsolicited event to notify the client that an open order has been updated by the Trading System. This message will be sent for the order quantities canceled because of Self Trade Prevention. This message is also sent to reinstate existing GoodForTime orders after the turn of a logical trading session for maintenance.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 8
37	OrderID	Y	Unique identifier for an Order assigned by the exchange.
11	ClOrdID	Y	The ClOrdID associated with the order.
17	ExecID	Y	Unique identifier of execution message as assigned by the exchange.



Tag	Field Name	Req'd	Meaning/Values
21025	CorrelationID	Y	A reference number used to match this order with orders broadcast on MEMOIR Depth for Crypto. Values will be unique per token, per trading session.
150	ExecType	Y	D - Restated
39	OrdStatus	Y	0 - New 1 - Partially filled 4 - Canceled
55	Symbol	Y	As stated in the order.
54	Side	Y	As stated in the order.
21023	QuoteIndex	N	As stated in NewOrderBulk request or 0 for NewOrderSingle request.
151	LeavesQty	Y	Quantity open for execution. See Order Quantities for more details.
14	CumQty	Y	Total quantity filled on the order. See Order Quantities for more details.
44	Price	N	As stated in the order.
38	OrderQty	Y	The quantity after the restatement. See Order Quantities for more details.
31	LastPx	Y	For ExecRestatementReason (5): The match price if this order had not been prevented from executing due to STP restrictions.
32	LastQty	N	For ExecRestatementReason (5): The quantity that would have matched if this order had not been prevented from executing due to STP restrictions.
378	ExecRestatementReason	Y	1 - GTT renewal / restatement. GTT order is reinstated on the book following a maintenance window. 5 - Partial decline of OrderQty (38)
1	Account	Y	Account associated to the order.
453	NoPartyIDs	Y	Number of PartyID's to follow (Always 1).
447	PartyIDSource	Y	Identifies class or source of the PartyID (448) value.
448	PartyID	Y	CPID as stated in the order, or the default CPID for the Account if not specified in the order.
452	PartyRole	Y	Identifies the type or role of the PartyID (448) specified.
60	TransactTime	Y	UTC timestamp since epoch with millisecond precision.
583	LnkID	N	As stated in the order.
	Standard Trailer	Y	

### 10.3 UserNotification

The UserNotification message is used to notify the user of an event or information about the exchange. This message will be used to provide a logical session shutdown warning due to an up and coming maintenance window.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = CB
926	UserStatus	Y	Reason for notification.
	Standard Trailer	Y	