



FIX Specification: Order Entry and Drop Copy Protocols

FIX Application Layer and Business Messages

Contents

0.1	Revision History	2
1	Overview	3
1.1	FIX Session Protocol and Message Framing	3
1.2	TXSE Custom Fields	3
1.3	Drop Copy	4
1.4	Pegged Orders	4
1.5	Display Price Slide Instructions	5
1.6	Market Orders	6
1.7	Hidden Orders	6
1.8	Reserve Orders	7
1.9	Post-Only Orders	8
1.10	Good-'Til-Time Orders	8
1.11	Self Match Prevention	9
1.12	Mass Cancel Behavior	10
1.13	Crossed Market Conditions	11
2	Types	12
2.1	CancelReason	12
2.2	CxlRejReason	12
2.3	CxlRejResponseTo	13
2.4	ExecInst	14
2.5	ExecRestatementReason	14
2.6	ExecType	14
2.7	LastLiquidityInd	14
2.8	OrdRejReason	14
2.9	OrdStatus	16
2.10	OrdType	16
2.11	OrderCapacity	16
2.12	PriceSlideInstruction	16

2.13 SelfMatchPreventionInstruction	17
2.14 SelfMatchScope	17
2.15 Side	17
2.16 TimeInForce	18
2.17 TradeLiquidityIndicator	18
3 FIX Application Messages from Member to Exchange	19
3.1 NewLimitOrderSingle	19
3.2 NewMarketOrderSingle	20
3.3 OrderCancelRequest	20
3.4 OrderReplaceRequest	21
4 FIX Application Messages from Exchange to Member	23
4.1 ExecutionReportNew	23
4.2 ExecutionReportRejected	24
4.3 ExecutionReportCanceled	25
4.4 OrderCancelReject	26
4.5 ExecutionReportReplaced	26
4.6 OrderReplaceReject	27
4.7 ExecutionReportTrade	28
4.8 ExecutionReportReplenished	29
4.9 ExecutionReportRepriced	29

0.1 Revision History

Date	Version	Notes
July 23, 2025	0.1	Initial version. Known gaps: Risk Controls, Liquidity code and Symbology.
September 09, 2025	0.6.draft	Alpha release. Refined message layout, identifiers and enumerations in line with early internal feedback.
October 28, 2025	0.7.draft	Updated support contact email.
November 04, 2025	0.8.draft	Updated TimeInForce documentation.
December 10, 2025	0.9.draft	Simplified session protocol section to reference comprehensive Session Protocol document.
December 16, 2025	0.10.draft	ClOrdId and OrigClOrdId support up to 20 characters.
January 05, 2026	0.11.draft	Added LastLiquidityInd (tag 851) and TradeLiquidityIndicator (tag 9730) to ExecutionReportTrade.
January 08, 2026	0.12.draft	Drop Copy now supports all messages from Exchange to Member, with OnBehalfOfComplID added to each message.
January 16, 2026	0.13.draft	Added SelfMatchScope, SelfMatchPreventionInstruction, PriceSlideInstruction, and PegOffsetValue to Replace requests.
January 20, 2026	0.14.draft	Added support for Regulation SHO and extended related enumerations.

1 Overview

The **Texas Stock Exchange (TXSE)** supports electronic order entry and market data dissemination using the Financial Information eXchange (FIX) protocol, an industry standard for real-time electronic communication in the global financial markets. The Exchange adheres to FIX Protocol version 5.0 Service Pack 2 (FIX 5.0 SP2), as defined and maintained by the FIX Trading Community (<https://www.fixtrading.org>).

This specification outlines the Exchange's implementation of FIX 5.0 SP2 Application Layer for Order Entry and Drop Copy Protocols.

The document is intended for use by firms integrating with the Exchange via FIX, including broker-dealers, trading system vendors, and market participants developing or certifying FIX engines or client applications. Adherence to the Exchange FIX specification ensures consistent and reliable interaction with the Exchange's trading platform.

1.1 FIX Session Protocol and Message Framing

For comprehensive documentation on the FIX Session Layer, including session establishment, standard header and trailer field requirements, sequence number management, heartbeat mechanism, connection monitoring, session termination, error handling, and message validation, please refer to the **TXSE FIX Specification: Session Protocol** document.

The Session Protocol document provides detailed specifications for session-level protocol compliance, connectivity requirements, and message framing that apply to all FIX communications with the Exchange.

1.2 TXSE Custom Fields

TXSE extends the FIX 5.0 SP2 protocol with custom fields (tag numbers 8000-9999) to support exchange-specific functionality.

These custom fields enable TXSE-specific order handling, self-match prevention, price sliding for regulatory compliance, and other exchange-unique capabilities that are not part of the standard FIX protocol.

Tag	Field Name	Type	Notes
8000	PriceSlideInstruction	ENUM	Price slide instruction for compliance with Reg NMS Rule 610(d). Enum PriceSlideInstruction .
8001	SelfMatchScope	ENUM	Self-match prevention scope (by Member, MPID, or Member Group). Enum SelfMatchScope .
8002	OperationalHaltReason	ENUM	Operational reason for halt (administrative or technical). Enum OperationalHaltReason .
8003	CancelReason	ENUM	Reason for order cancellation. Enum CancelReason .
9000	LocateBroker	STRING	Used for short sale orders to identify the broker that has loaned the stock to settle the short sale. MPID should contain upper-case alpha only.
9001	MaxReplenishTimeRange	LONG	Defines the maximum range to be used when calculating a random refresh time, in nanoseconds.
9002	UserData	LONG	Pass-through field for use by clients. Not used by the Exchange.
9004	MemberGroup	STRING	Member-supplied grouping. Used for both self-match prevention and mass cancel, in conjunction with MPID or Sender-Comp.
9005	CancelAtEntryIfCrossed	BOOLEAN	Cancel the order at entry if market conditions are crossed.
9008	RankPrice	PRICE	Rank price, when different from limit price.

Enumerations The following subsections define enumeration types used in TXSE FIX messages. Custom TXSE enumeration values are marked as "(custom TXSE value)" in the Notes column.

1.3 Drop Copy

TXSE's Drop Copy service provides real-time updates to designated observer connections. Drop Copy uses the same message formats as Order Entry for all messages from Exchange to Member, with additional fields listed below.

Drop Copy Additional Fields

Tag	Field Name	Type	Required	Notes
115	OnBehalfOfComplD	STRING	N	Application name of the originating port

1.4 Pegged Orders

Pegged orders automatically price relative to the National Best Bid and Offer (NBBO) using the `referencePriceTarget` field, specified in basis points relative to the NBBO spread.

Pricing Calculation

$$P_{\text{buy}} = \text{NBB} + \left(\frac{\text{referencePriceTarget}}{10,000} \times (\text{NBO} - \text{NBB}) \right) \quad (1)$$

$$P_{\text{sell}} = \text{NBO} - \left(\frac{\text{referencePriceTarget}}{10,000} \times (\text{NBO} - \text{NBB}) \right) \quad (2)$$

Examples

BUY Order:

referencePriceTarget: 2,500 (25%)
 NBB: \$10.00, NBO: \$11.00 (spread: \$1.00)
 Order Price: \$10.00 + (2,500/10,000 × \$1.00) = \$10.25

SELL Order:

referencePriceTarget: 7,000 (70%)
 NBB: \$10.00, NBO: \$11.00 (spread: \$1.00)
 Order Price: \$11.00 - (7,000/10,000 × \$1.00) = \$10.30

Common Peg Types

Standard peg types can be implemented using specific `referencePriceTarget` values:

- **PRIMARY Peg:** `referencePriceTarget` = 0 (0%) - Orders price at NBB (buy) or NBO (sell)
- **MID Peg:** `referencePriceTarget` = 5,000 (50%) - Orders price at NBBO midpoint
- **MARKET Peg:** `referencePriceTarget` = 10,000 (100%) - Orders price at NBO (buy) or NBB (sell)

Execution Restrictions

Pegged orders will not execute in locked, crossed, one-sided, or no-sided markets where normal NBBO pricing relationships do not exist.

Priority and Price Rounding

When different referencePriceTarget values calculate to the same final price due to rounding, arrival time determines priority.

Narrow Spread Example: Both orders price at \$10.02, so if the 70% order arrived first, it has priority.

NBB: \$10.00, NBO: \$10.03, Spread: \$0.03
 70% BUY: $\$10.00 + (0.70 \times \$0.03) = \$10.021 \rightarrow \10.02
 90% BUY: $\$10.00 + (0.90 \times \$0.03) = \$10.027 \rightarrow \10.02

Wide Spread Example: Different final prices result in standard price-time priority as 90% BUY has better price.

NBB: \$10.00, NBO: \$10.10, Spread: \$0.10
 70% BUY: $\$10.00 + (0.70 \times \$0.10) = \$10.07$
 90% BUY: $\$10.00 + (0.90 \times \$0.10) = \$10.09$

Midpoint Order with a 50% referencePriceTarget will trade at midpoint of the NBBO, which may result in a sub-penny execution price for stocks over \$1.00. Midpoint executions might also apply to other peg targets depending on the spread at the time of execution. For example, for a stock with an NBBO greater than \$1.00 and a spread of \$0.01, all referencePriceTargets greater than or equal to 50% and less than 100% are all effectively midpoint pegs, until the spread changes.

Sub \$1.00 stocks The same peg math applies to stocks under \$1.00, but valid peg prices are floored to an MPV of \$0.0001

Peg Reference Price Target Rounding While the Member is free to enter any value for the referencePriceTarget, the Exchange may floor the peg target to a minimum variation for a given stock. For example, the Exchange may determine that for a specific stock, only increments of 10% are economically meaningful. If the Member enters a referencePriceTarget of 23%, the Exchange may floor it to 20%. The Member will see the result on the order acknowledgment. The Exchange reserves the right to change the minimum peg target increment at any time for any stock. Values of 0%, 50% and 100% will never be altered.

1.5 Display Price Slide Instructions

Display Price Slide Instructions automatically adjust order display prices to comply with Regulation NMS Rule 610(d), preventing locked or crossed markets. When an order would lock (match) or cross (trade through) the Reg NMS markets, the Exchange will reprice the displayed price while preserving the original limit price for execution.

Price Slide Behavior

When sliding occurs:

- **Internal ranking:** Order is ranked at the locking price for priority purposes
- **Market display:** Order is displayed one tick away
- **Original limit price:** Retained for execution purposes

Message Flow

Price sliding generates the following message sequence:

- LimitOrderAccepted: Acknowledges order with price field set to original limit price
- OrderRestated: Reports the ranked price with reason = REPRICED when sliding occurs

Examples

Locking Scenario:

Market: AAPL \$150.00 x \$150.01

Incoming order: Buy 1000 AAPL @ \$150.01 (would lock the offer)

- NO_PRICE_SLIDE: Order canceled
- SINGLE_PRICE_SLIDE_ON_LOCK_AND_CROSS: Ranked at \$150.01, Displayed at \$150.00
- MULTIPLE_PRICE_SLIDES_ON_LOCK_AND_CROSS: Ranked at \$150.01, Displayed at \$150.00

Crossing Scenario:

Market: AAPL \$150.00 x \$150.01

Incoming order: Buy 1000 AAPL @ \$150.02 (would cross the offer)

- NO_PRICE_SLIDE: Order canceled
- SINGLE_PRICE_SLIDE_ON_LOCK_AND_CROSS: Ranked at \$150.01, Displayed at \$150.00
- MULTIPLE_PRICE_SLIDES_ON_LOCK_AND_CROSS: Ranked at \$150.01, Displayed at \$150.00
- SINGLE_PRICE_SLIDE_LOCK_ONLY: Order canceled (crossing not allowed)
- MULTIPLE_PRICE_SLIDES_LOCK_ONLY: Order canceled (crossing not allowed)

Priority Impact

Each reprice creates a new timestamp and the order loses time priority at its new price level. Orders retain relative priority among other repriced orders based on original entry time.

1.6 Market Orders

Market orders provide a convenient way to execute immediate trades without specifying a price. Internally, they are handled as limit orders with a 100% peg to the reference price, ensuring aggressive pricing (maximum price for buy orders, minimum price for sell orders). Market orders are configured with no resting capability and will immediately cancel any unfilled portion after execution attempts, guaranteeing they never rest on the book.

1.7 Hidden Orders

Hidden orders do not have Price Slide Instructions; however, the Exchange will collar the limit price. If at order entry the order would cross the NBO (for BUY) or NBB (for SELL), the Exchange will adjust the order's limit price to the locking price. This adjustment happens only once at time of entry and the order is **NOT** repriced subsequently.

Example

Market: AAPL \$150.00 x \$150.01
 Incoming order: Buy Hidden 1000 AAPL @ \$150.10 (would cross the offer)
 Order accepted: Buy Hidden 1000 AAPL @ \$150.01 (price adjusted to lock)

Note: The order acknowledgment message will reflect the adjusted limit price.

1.8 Reserve Orders

Reserve orders utilize a "Reserve Quantity" - the non-displayed portion of an order while a portion is displayed on the TXSE Book. Both the displayed portion and the Reserve Quantity are available for execution.

Basic Behavior

Reserve orders are created by setting `isHidden = true` and specifying a `maxFloorQty` that is less than the total `orderQty`. The Exchange displays a portion while keeping the remaining quantity (Reserve Quantity) hidden.

Replenishment occurs when the displayed portion is reduced to less than a Round Lot (typically 100 shares). Each replenishment creates a new `OrderId` while preserving the original `ClOrdId`.

Replenishment Types

- **Fixed Replenishment:** Default
 - Quantity: Replenishes to `maxFloorQty`
 - Time: Immediate replenishment
- **Random Quantity Replenishment:** Replenishment quantities are randomly selected within a range in Round Lot increments only
 - Range: (`maxFloorQty - maxReplenishQtyRange`) to (`maxFloorQty + maxReplenishQtyRange`)
 - Example: `MaxFloor = 500`, `Range = 200` → Possible quantities: [300, 400, 500, 600, 700] shares
- **Random Time Replenishment:** Replenishment occurs after a random delay
 - Range: 0 to `maxReplenishTimeRange`
 - Example: `MaxReplenishTimeRange=500us` → Random delay between 0-500us after execution

Note: Random Quantity and Random Time can be configured together on the same order.

Active Replenishments

At any point in time, a reserve order can have:

- 0 replenishments: During time delay periods (Random Time configuration) or transient system states due to non-atomic replenishment processing
- 1 replenishment: Normal use case with active displayed quantity
- 2 replenishments: When the first replenishment falls below Round Lot size, triggering a second replenishment while the first may still have displayed leaves quantity

Order Restatement Messages

Members receive order restatement messages for each replenishment. The restatement will indicate:

- New replenishment OrderId
- Restatement reason: RESERVE_REPLENISHED
- New displayed quantity

Key Technical Details

- Each replenishment creates a new OrderId but preserves the original CIOld
- Exchange obfuscates OrderIds on data feeds for replenishment orders
- Reserve Quantity remains fully executable during replenishment delays
- Random replenishment quantities must be in Round Lot increments
- If remainder is less than replenishment amount, entire remainder is displayed
- Order management operations use the original CIOld

1.9 Post-Only Orders

Post-Only orders are designed to add liquidity to the order book without removing existing liquidity. These orders will only post to the book as resting orders and will never execute immediately against existing orders.

Behavior

When a Post-Only order is submitted:

- If marketable without Display Price Slide Instructions: Order is canceled (would immediately execute against existing orders)
- If marketable with Display Price Slide Instructions: Order may be repriced to avoid immediate execution, depending on the slide instruction
- If not marketable: Order posts to the book at the specified limit price

Key Characteristics

- Liquidity provision only: Guarantees the order will either rest on the book or be canceled
- No immediate execution: Eliminates the risk of unexpected immediate fills
- Works with Display Price Slide Instructions: Can be combined with price slide instructions to automatically adjust marketable orders instead of canceling them

1.10 Good-'Til-Time Orders

Good-'Til-Time (GTT) orders allow members to specify a custom expiration time using the `expireTime` field. Orders with a GTT `TimeInForce` will remain active on the Exchange until the specified time or the end of the trading day is reached, at which point any remaining quantity will be automatically canceled.

Minimum Expiration Time

The Exchange enforces a minimum expiration threshold of 1 millisecond from the current time. Orders submitted with an `expireTime` less than the current time plus 1 millisecond will be rejected with reason `INVALID_EXPIRE_TIME`.

Trading Session Lifetime Limit

GTT order lifetimes are limited to the current trading day and will not persist beyond the end of the After Hours Session. If a Member specifies an `expireTime` that extends beyond the end of the After Hours Session, the Exchange will accept the order but will automatically cancel any remaining quantity at the end of the After Hours Session rather than rejecting the order at entry. This ensures orders do not inadvertently remain active across trading days while providing members the convenience of not having to calculate exact session end times.

Session End Behavior

- Orders with `expireTime` before the After Hours Session end: Canceled at the specified `expireTime`
- Orders with `expireTime` after the After Hours Session end: Canceled at the After Hours Session end
- Cancel reason in both cases: `RELATED_TO_TIME_IN_FORCE`

1.11 Self Match Prevention

TXSE's Self Match Prevention functionality prevents orders from the same trading entity from matching against each other, supporting regulatory compliance and risk management requirements consistent with the TXSE Rule 11.007(d).

The system operates across configurable scopes including by Member (`BY_MEMBER`), by Market Participant Identifier (`BY_MPID`), by a combination of both Member and Member Group (`BY_MEMBER_GROUP`), or a combination of both MPID and Member Group (`BY_MPID_AND_MEMBER_GROUP`).

Self-match prevention settings may be configured at the port level as default parameters for all orders submitted through that connection. These port-level defaults can be overridden on a per-order basis through the optional `selfMatchScope` and `selfMatchInstruction` fields, providing members with both operational efficiency and granular control over their order interaction policies.

When a potential self-match is detected between orders, the Exchange applies the specified prevention instruction: cancel the incoming order (`CANCEL_NEWEST`), cancel the resting order (`CANCEL_OLDEST`), cancel both orders entirely (`CANCEL_BOTH`), cancel only the smaller order if sizes differ or both if equal (`CANCEL_SMALLEST`), or cancel the overlapping portion while preserving the remainder of the larger order (`DECREMENT_AND_CANCEL`).

IMPORTANT: Only the aggressive order's self-match prevention instructions are considered during a match. The resting order's instructions are ignored.

SelfMatchPrevented Message

Orders affected by self-match prevention will generate a `SelfMatchPrevented` message for each order involved. This message reports:

- `execPrice`: The price at which the match would have occurred
- `execQty`: The shares that would have executed
- `canceledQty`: The number of shares decremented or canceled from the order
- `leavesQty`: The remaining quantity still active on the book after the prevention action

For `DECREMENT_AND_CANCEL` instructions, the `canceledQty` reflects only the overlapping portion that was removed, while `leavesQty` indicates any remaining shares that continue to rest on the book. For full cancellation instructions like `CANCEL_NEWEST`, `CANCEL_OLDEST`, or `CANCEL_BOTH`, the `canceledQty` will equal the remaining open quantity and `leavesQty` will be 0.

1.12 Mass Cancel Behavior

Cancel a large group of orders based on the Member Owned scope.

Member Owned Scope

- BY_MEMBER_OWNED_SENDER_COMPS: Scope mass cancels to all the ports owned by the Member
- BY_MEMBER_OWNED_MPIDS: Scope mass cancels to all the MPIDs owned by the Member

Optional Filter Fields

The optional filter fields are logically ANDed together. Each additional field narrows the number of orders affected.

The ClOrdId must be paired with SenderComp, and will cancel the single specific order from the order entry port associated with the SenderComp.

Canceling of orders is done on a best-effort basis. It is not atomic. The OrderCanceled messages will flow back to the order entry ports that entered the orders, and any drop copy ports that are following the orders. They will not flow back to the order entry port that sent the MassCancel message, except where orders entered by the same port are affected.

Examples

Mass Cancel on all Member Owned Ports

Mass cancel all orders entered on ports owned by the Member, but **NOT** orders entered on behalf of the Member via service bureau owned ports.

Message Type: 'V' (0x56)
scope: BY_MEMBER_OWNED_SENDER_COMPS

Mass Cancel on all Member Owned MPIDs

Mass cancel all orders associated with the Member, across all member owned ports **AND** service bureau owned ports (on-behalf-of orders).

Message Type: 'V' (0x56)
scope: BY_MEMBER_OWNED_MPIDS

Mass Cancel a specific MPID by a Member

Mass cancel all orders associated with the MPID, across all member owned ports **AND** service-bureau owned ports (on-behalf-of orders).

Message Type: 'V' (0x56)
scope: BY_MEMBER_OWNED_MPIDS
mpid: ABCD

Mass Cancel by MPID (Service Bureau Initiated)

Allows a Service Bureau to mass cancel all orders for a specified MPID, limited exclusively to orders submitted on behalf of that MPID through Service Bureau-owned ports.

Message Type: 'V' (0x56)
scope: BY_MEMBER_OWNED_SENDER_COMPS
mpid: ABCD

1.13 Crossed Market Conditions

When a Protected Bid crosses a Protected Offer, the Exchange restricts executions to prevent trades at unreasonable prices, per Regulation NMS Rule 610(e).

Execution Price Restrictions During Crossed Markets

Per Rule 11.009(a)(2), during crossed markets the Exchange will not execute:

- Buy orders at prices more than 0.5% (or \$0.05, whichever is greater) above the lowest Protected Offer
- Sell orders at prices more than 0.5% (or \$0.05, whichever is greater) below the highest Protected Bid

Cancel on Crossed Market Instruction

Members may include a `cancelAtEntryIfCrossed` instruction on orders. When set, the Exchange will immediately cancel the incoming order if a Protected Bid is crossing a Protected Offer at time of entry, rather than attempting execution within the price restrictions.

Order Handling During Crossed Markets

- Limit Orders will follow Display Price Slide Instructions. Refer to [documentation](#) for details.
- Market Orders will not execute when a locked or crossed market exists.
- Pegged Orders will not execute when a locked or crossed market exists.

Examples

Crossed Market - Buy Order Execution Cap:

NBBO: \$10.05 x \$10.03 (crossed)

Max Buy Execution Price: $\$10.03 + \max(\$0.05, \$10.03 \times 0.005) = \10.08

Incoming: Buy 1000 @ \$10.10

Result: Can execute up to \$10.08; orders/liquidity above \$10.08 unavailable

Crossed Market - With Cancel Instruction:

NBBO: \$10.05 x \$10.03 (crossed)

Incoming: Buy 1000 @ \$10.10 with `cancelAtEntryIfCrossed`

Result: Order immediately canceled (no execution attempt)

High-Priced Stock Example:

NBBO: \$1,005 x \$1,000 (crossed)

Max Buy Price: $\$1,000 + \max(\$0.05, \$5.00) = \$1,005$

Incoming: Buy 100 @ \$1,010 with `cancelAtEntryIfCrossed =false`

Result: Can execute up to \$1,005; cannot access liquidity above \$1,005

ISO Exception

Orders marked as Intermarket Sweep Orders (ISO) may execute beyond these price restrictions, as the sender assumes responsibility for simultaneously routing orders to protected quotations.

2 Types

Enum types used in this protocol. Enums that do not deviate from the standard, or whose meanings are commonly known, are not listed here.

2.1 CancelReason

Reason that an order was canceled. (custom TXSE enumeration)

Name	Value	Notes
REQUESTED_BY_USER	'1'	Cancel due to request by client.
RELATED_TO_TIME_IN_FORCE	'2'	Cancel due to time in force expiration.
RELATED_TO_MIN_QTY	'3'	Cancel due to inability to meet minimum quantity.
REG_NMS_VIOLATION_NO_SLIDE	'4'	Canceled as order would violate Regulation NMS Rule 610(d) by creating a locked or crossed market, and price sliding was either not selected or not applicable.
MARKETABLE_RESERVE	'5'	Reserve orders will be canceled when deemed marketable at another exchange either at new order entry or replenishment.
SELF_MATCH_PREVENTION	'6'	Cancel due to self-match prevention instructions.
REPLENISHMENT_CANCELED_DUE_TO_RESERVE	'7'	Replenishment canceled related to the Reserve order being canceled or replaced.
RELATED_TO_ORDER_TYPE	'8'	Cancel due to order type e.g. Market Orders do not rest.
CANCELED_DUE_TO_CROSSED_MARKETS	'9'	Cancel due to client instruction on crossed markets.
CANCELED_DUE_TO_MASS_CANCEL_REQUEST	'10'	Cancel due to a mass cancel request.
EXCHANGE_LOCKED_OR_CROSSED_NO_SLIDE	'11'	Canceled as order would lock or cross the Exchange book, and price sliding was either not selected or not applicable.
REG_SHO_VIOLATION_NO_SLIDE	'12'	Canceled as order would violate Regulation SHO Rule 201 by display at or below the National Best Bid during a short sale price test restriction, and price sliding was either not selected or not applicable

2.2 CxlRejReason

Indicates the reason that a customer action was rejected by the exchange. Applicable to both cancel and replace requests.

Name	Value	Notes
TOO_LATE_TO_CANCEL	'0'	Attempt to modify, replace, or cancel an order that has already canceled or fully executed.
UNKNOWN_ORDER	'1'	Attempt to modify or replace a client order that was not found for the current session on this port.

Name	Value	Notes
DUPLICATE_CLORDID_RECEIVED	'6'	Client Order Id has already been used for the current session on this port.
PRICE_EXCEEDS_CURRENT_PRICE_BAND	'8'	Invalid limit price.
INVALID_REFERENCE_PRICE_TARGET	'100'	Invalid reference price target related to the order attributes. (custom TXSE value)
INVALID_SIDE	'103'	Invalid side. (custom TXSE value)
INVALID_MAX_FLOOR_QUANTITY	'104'	Invalid max floor quantity related to the order attributes. (custom TXSE value)
INVALID_LOCATE_REQUIRED_FLAG	'108'	Invalid locate-required flag related to the order attributes. (custom TXSE value)
INVALID_MPID	'110'	Invalid market participant identifier. (custom TXSE value)
INVALID_IS_POST_ONLY_FLAG	'111'	Invalid 'Is Post Only' flag related to the order attributes. (custom TXSE value)
ORDER_INVALID_FOR_TRADING_SESSION	'113'	Order not permitted during the current market session. (custom TXSE value)
ORDER_INVALID_FOR_TRADING_STATUS	'114'	Exchange is not currently accepting orders. (custom TXSE value)
ORDER_INVALID_FOR_SYMBOL_STATUS	'115'	Symbol is not currently accepting orders. (custom TXSE value)
INVALID_IS_ISO_FLAG	'116'	Invalid 'Is ISO' flag related to the order attributes (ISO = Intermarket Sweep Order). (custom TXSE value)
TRADING_DISABLED_FOR_PORT_OR_MPID	'117'	Trading is disabled for this port or MPID based on port settings configuration. (custom TXSE value)
TRADING_DISABLED_FOR_NON_TEST_SYMBOLS	'118'	Trading is restricted to test symbols only based on port settings configuration. (custom TXSE value)
TRADING_DISABLED_FOR_ISO	'120'	ISO (Intermarket Sweep Order) trading is disabled based on port settings configuration. (custom TXSE value)
MAXIMUM_ORDER_QUANTITY_BREACHED	'121'	Order quantity exceeds the maximum shares limit configured in port settings. (custom TXSE value)
MAXIMUM_PRICE_BREACHED	'122'	Limit price exceeds the maximum acceptable price for the Exchange. (custom TXSE value)
UNKNOWN_SYMBOL	'200'	Symbol ID is not valid for the current session. (custom TXSE value)
INCORRECT_QUANTITY	'201'	Invalid order quantity. (custom TXSE value)
EXCEEDED_MAX_NOTIONAL_ORDER_AMT	'202'	Order notional value exceeds the maximum notional limit configured in port settings. (custom TXSE value)
MODIFICATION_NOT_PERMITTED	'203'	Order modification not permitted. (custom TXSE value)

2.3 CxlRejResponseTo

Name	Value	Notes
ORDER_CANCEL_REQUEST	'1'	
ORDER_CANCEL_REPLACE_REQUEST	'2'	

2.4 ExecInst

Name	Value	Notes
PARTICIPATE_DO_NOT_INITIATE	'6'	Indicates a Post Only order instruction.
INTERMARKET_SWEEP	'f'	Indicates a Reg NMS Inter-market sweep order instruction.

2.5 ExecRestatementReason

Reason for order restatement events triggered by the Exchange.

Name	Value	Notes
REPRICING_OF_ORDER	'3'	A reprice eligible order has been repriced to a new price.
RESERVE_REPLENISHED	'100'	A new displayed slice of a reserve order has been added to the displayed book. (custom TXSE value)

2.6 ExecType

Name	Value	Notes
NEW	'0'	
CANCELED	'4'	
REPLACED	'5'	
REJECTED	'8'	
RESTATED	'D'	
TRADE	'F'	

2.7 LastLiquidityInd

Indicator to identify whether this fill was a result of a liquidity provider providing or liquidity taker taking the liquidity, per FIX 5.0 specifications.

Name	Value	Notes
ADDED_LIQUIDITY	'1'	Added Liquidity.
REMOVED_LIQUIDITY	'2'	Removed Liquidity.

2.8 OrdRejReason

Indicates the reason that a customer action was rejected by the exchange.

Name	Value	Notes
UNKNOWN_SYMBOL	'1'	Symbol ID is not valid for the current session.
UNKNOWN_ORDER	'5'	Client Order Id is invalid.
DUPLICATE_ORDER	'6'	Client Order Id has already been used for the current session on this port.

Name	Value	Notes
STALE_ORDER	'8'	Attempt to modify, replace, or cancel an order that has already canceled or fully executed.
INCORRECT_QUANTITY	'13'	Invalid order quantity.
PRICE_EXCEEDS_CURRENT_PRICE_BAND	'16'	Invalid limit price.
EXCEEDED_MAX_NOTIONAL_ORDER_AMT	'27'	Order notional value exceeds the maximum notional limit configured in port settings.
INVALID_REFERENCE_PRICE_TARGET	'100'	Invalid reference price target related to the order attributes. (custom TXSE value)
INVALID_IS_HIDDEN_FLAG	'101'	Invalid 'Is Hidden' flag related to the order attributes. (custom TXSE value)
INVALID_ORDER_TYPE	'102'	Invalid order type related to the order attributes. (custom TXSE value)
INVALID_SIDE	'103'	Invalid side. (custom TXSE value)
INVALID_MAX_FLOOR_QUANTITY	'104'	Invalid max floor quantity related to the order attributes. (custom TXSE value)
INVALID_MAX_REPLENISH_QUANTITY_RANGE	'105'	Invalid maximum replenish quantity range. (custom TXSE value)
INVALID_MAX_REPLENISH_TIME_RANGE	'106'	Invalid maximum replenish time range. (custom TXSE value)
INVALID_MINIMUM_QUANTITY	'107'	Invalid minimum quantity. (custom TXSE value)
INVALID_LOCATE_REQUIRED_FLAG	'108'	Invalid locate-required flag related to the order attributes. (custom TXSE value)
INVALID_TIME_IN_FORCE	'109'	Invalid time in force. (custom TXSE value)
INVALID_MPID	'110'	Invalid market participant identifier. (custom TXSE value)
INVALID_IS_POST_ONLY_FLAG	'111'	Invalid 'Is Post Only' flag related to the order attributes. (custom TXSE value)
INVALID_EXPIRE_TIME	'112'	Invalid expire time. (custom TXSE value)
ORDER_INVALID_FOR_TRADING_SESSION	'113'	Order not permitted during the current market session. (custom TXSE value)
ORDER_INVALID_FOR_TRADING_STATUS	'114'	Exchange is not currently accepting orders. (custom TXSE value)
ORDER_INVALID_FOR_SYMBOL_STATUS	'115'	Symbol is not currently accepting orders. (custom TXSE value)
INVALID_IS_ISO_FLAG	'116'	Invalid 'Is ISO' flag related to the order attributes (ISO = Intermarket Sweep Order). (custom TXSE value)
TRADING_DISABLED_FOR_PORT_OR_MPID	'117'	Trading is disabled for this port or MPID based on port settings configuration. (custom TXSE value)
TRADING_DISABLED_FOR_NON_TEST_SYMBOLS	'118'	Trading is restricted to test symbols only based on port settings configuration. (custom TXSE value)
TRADING_DISABLED_FOR_ORDER_CAPACITY	'119'	Trading is disabled for the specified order capacity based on port settings configuration. (custom TXSE value)
TRADING_DISABLED_FOR_ISO	'120'	ISO (Intermarket Sweep Order) trading is disabled based on port settings configuration. (custom TXSE value)

Name	Value	Notes
MAXIMUM_ORDER_QUANTITY_BREACHED	'121'	Order quantity exceeds the maximum shares limit configured in port settings. (custom TXSE value)
MAXIMUM_PRICE_BREACHED	'122'	Limit price exceeds the maximum acceptable price for the Exchange. (custom TXSE value)

2.9 OrdStatus

Name	Value	Notes
NEW	'0'	
PARTIALLY_FILLED	'1'	
FILLED	'2'	
CANCELED	'4'	
REPLACED	'5'	
REJECTED	'8'	

2.10 OrdType

Name	Value	Notes
MARKET	'1'	
LIMIT	'2'	

2.11 OrderCapacity

Broker capacity in sending order.

Name	Value	Notes
AGENCY	'A'	Acting on behalf of clients.
PRINCIPAL	'P'	Acting on behalf of Broker-Dealer.
RISKLESS_PRINCIPAL	'R'	Order to offset client order.

2.12 PriceSlideInstruction

For additional information, refer to Display Price Slide Instructions [documentation](#). (custom TXSE enumeration)

Name	Value	Notes
NO_PRICE_SLIDE	'0'	No Display-Price Slide instruction. Orders that would create locking or crossing quotations will be canceled.
SINGLE_PRICE_SLIDE_ON_LOCK_AND_CROSS	'1'	Single price slide applied to both locking and crossing quotations. Adjusts order price only upon entry and one additional time following an NBBO change.

Name	Value	Notes
MULTIPLE_PRICE_SLIDES_ON_LOCK_AND_CROSS	'2'	Multiple price slides applied to both locking and crossing quotations. Continuously re-ranks and re-displays orders as NBBO changes throughout the order's life.
SINGLE_PRICE_SLIDE_LOCK_ONLY	'3'	Single price slide applied only to locking quotations. Crossing quotations will be canceled. Note: Given this only serves locked quotations, a multiple price slide option could only slide once, thus making it redundant.

2.13 SelfMatchPreventionInstruction

Determines the behavior when orders from the same entity would match. For additional information, refer to Self Match Prevention [documentation](#).

Name	Value	Notes
CANCEL_AGGRESSIVE	'1'	Cancels the incoming order. Keeps the resting order on the book.
CANCEL_PASSIVE	'2'	Cancels the resting order. Keeps the incoming order on the book.
CANCEL_AGGRESSIVE_PASSIVE	'3'	Cancels both orders entirely regardless of size.
NO_SELF_MATCH_PREVENTION	'100'	Self-match prevention is disabled. (custom TXSE value)
CANCEL_SMALLEST	'101'	If orders are equal size: cancels both. If different sizes: cancels the smaller order, keeps the larger one. (custom TXSE value)
DECREMENT_AND_CANCEL	'102'	If orders are equal size: cancels both. If different sizes: cancels the overlapping portion, keeps the remainder of the larger order. (custom TXSE value)

2.14 SelfMatchScope

Determines the scope for self-match prevention. For additional information, refer to Self Match Prevention [documentation](#). (custom TXSE enumeration)

Name	Value	Notes
BY_MEMBER	'0'	Match prevention applies to all orders from the same Member.
BY_MPID	'1'	Match prevention applies to orders from the same Member with the same MPID.
BY_MEMBER_GROUP	'2'	Match prevention applies to orders from the same Member within the same Member Group.
BY_MPID_AND_MEMBER_GROUP	'3'	Match prevention applies to orders from the same Member with the same MPID and Member Group.

2.15 Side

Determines direction of a trading action, including short-sell indicators.

Name	Value	Notes
BUY	'1'	Order is a buy order.
SELL	'2'	Sell order of existing inventory.
SELL_SHORT	'5'	Sell order of borrowed shares.
SELL_SHORT_EXEMPT	'6'	Sell order of borrowed shares exempt from certain regulations of Regulation SHO.

2.16 TimeInForce

Determines time when an order goes live and when it will be canceled. The Exchange has three trading sessions: Early Session, Regular Session and After Hours Session. The Regular Session is the "normal" trading day, typically between 9:30 ET and 16:00 ET. The Early and After Hours Sessions are bounded by the Regular Session and the Exchange's start and end times.

Name	Value	Notes
DAY	'0'	Accepted during Early and Regular Session. Order becomes live immediately upon entry and is canceled at the end of the Regular Session. Will be rejected during the After Hours Session.
IMMEDIATE_OR_CANCEL	'3'	Similar to SYS, but canceled immediately, after trading with resting liquidity on the Exchange.
GOOD_TILL_DATE	'6'	Similar to SYS, but canceled after the requested time has passed or the end of the After Hours Session, whichever comes first. See <code>expireTime</code> field in <code>LimitOrder</code> . For additional information, refer to GTT documentation .
SYSTEM	'S'	Accepted during any session. Live immediately upon entry. Canceled at the end of the After Hours Session. (custom TXSE value)
REGULAR_HOURS_ONLY	'R'	Accepted during the Regular Session only. Order becomes live immediately upon entry and is canceled at the end of the Regular Session. Will be rejected during the Early and After Hours Sessions. (custom TXSE value)

2.17 TradeLiquidityIndicator

Liquidity indicator codes for executions.

Name	Value	Notes
REMOVED_HIDDEN_LIQUIDITY	'0'	Removed hidden liquidity.
REMOVED_DISPLAYED_LIQUIDITY	'1'	Removed displayed liquidity.
ADDED_HIDDEN_LIQUIDITY	'2'	Added hidden liquidity.
ADDED_DISPLAYED_LIQUIDITY	'3'	Added displayed liquidity.

3 FIX Application Messages from Member to Exchange

Messages sent from the Member to the Exchange.

3.1 NewLimitOrderSingle

NewOrderSingle submitting a new limit order with a specified price.

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. 'D' for New Order Single.
60	TransactTime	Y	Time request was initiated. Required by FIX 5.0, but not used by the Exchange.
40	OrdType	Y	Enum OrdType . '2' for Limit.
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
55	Symbol	Y	Security identifier root represented in CMS format.
65	SymbolSfx	N	Security identifier suffix represented in CMS format. Defaults to blank if not provided.
54	Side	Y	Enum Side . Determines direction of a trading action, including short-sell indicators.
114	LocateReqd	N	True if a locate is required for short-sale, false otherwise.
9000	LocateBroker	N	Used for short sale orders to identify the broker that has loaned the stock to settle the short sale. MPID should contain upper-case alpha only.
44	Price	Y	Limit price. Must be greater than 0. Maximum price is \$100,000,000.00.
38	OrderQty	Y	Number of shares. Must be greater than 0.
59	TimeInForce	Y	Enum TimeInForce .
18	ExecInst	N	Enum ExecInst .
9005	CancelAtEntryIfCrossed	N	True to cancel the order at entry if market conditions are crossed.
528	OrderCapacity	Y	Enum OrderCapacity . Broker capacity in sending order.
109	ClientID	N	Market Participant Identifier. MPID should contain upper-case alpha only.
9004	MemberGroup	N	Member supplied grouping. Used for both self-match prevention and mass cancel, in conjunction with MPID or SenderComp.
8001	SelfMatchScope	N	Enum SelfMatchScope . Determines the scope for self-match prevention. For additional information, refer to Self Match Prevention documentation .
2964	SelfMatchPreventionInstruction	N	Enum SelfMatchPreventionInstruction . Determines the behavior when orders from the same entity would match. For additional information, refer to Self Match Prevention documentation .
8000	PriceSlideInstruction	N	Enum PriceSlideInstruction . For additional information, refer to Display Price Slide Instructions documentation .
211	PegOffsetValue	N	The target relative to the reference base price in basis points of the NBBO spread.
110	MinQty	N	Number of shares that must be executed to execute any shares.
1138	DisplayQty	N	Quantity to be displayed at one time with remaining quantity not displayed on the book.

Tag	Name	Req'd	Notes
1087	DisplayMinIncr	N	Defines the maximum range to be used when calculating a random refresh quantity.
9001	MaxReplenishTimeRange	N	Defines the maximum range to be used when calculating a random refresh time, in nanoseconds.
126	ExpireTime	N	Timestamp at which the order's open quantity will be canceled, valid only with GTT time in force. For additional information, refer to GTT documentation .
9002	UserData	N	Pass-through field for use by clients, not used by the exchange.

3.2 NewMarketOrderSingle

NewOrderSingle submitting a new market order.

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType . 'D' for New Order Single.
60	TransactTime	Y	Time request was initiated. Required by FIX 5.0, but not used by the Exchange.
40	OrdType	Y	Enum OrdType . '1' for Market.
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
55	Symbol	Y	Security identifier root represented in CMS format.
65	SymbolSfx	N	Security identifier suffix represented in CMS format. Defaults to blank if not provided.
54	Side	Y	Enum Side . Determines direction of a trading action, including short-sell indicators.
114	LocateReqd	N	True if a locate is required for short-sale, false otherwise.
9000	LocateBroker	N	Used for short sale orders to identify the broker that has loaned the stock to settle the short sale. MPID should contain upper-case alpha only.
38	OrderQty	Y	Number of shares. Must be greater than 0.
59	TimeInForce	Y	Enum TimeInForce .
528	OrderCapacity	Y	Enum OrderCapacity . Broker capacity in sending order.
109	ClientID	N	Market Participant Identifier. MPID should contain upper-case alpha only.
9004	MemberGroup	N	Member supplied grouping. Used for both self-match prevention and mass cancel, in conjunction with MPID or SenderComp.
8001	SelfMatchScope	N	Enum SelfMatchScope . Determines the scope for self-match prevention. For additional information, refer to Self Match Prevention documentation .
2964	SelfMatchPreventionInstruction	N	Enum SelfMatchPreventionInstruction . Determines the behavior when orders from the same entity would match. For additional information, refer to Self Match Prevention documentation .
9002	UserData	N	Pass-through field for use by clients, not used by the exchange.

3.3 OrderCancelRequest

OrderCancelRequest canceling all outstanding shares of an existing order.

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. 'F' for Order Cancel Request.
60	TransactTime	Y	Time request was initiated. Required by FIX 5.0, but not used by the Exchange.
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
41	OrigClOrdID	Y	Original ClOrdID. Must reference an existing order.
55	Symbol	Y	Security identifier root represented in CMS format. This field must not change from the value sent in the original order.
65	SymbolSfx	N	Security identifier suffix represented in CMS format. This field must not change from the value sent in the original order. Defaults to blank if not provided.
54	Side	Y	Enum Side . This field must match the value sent in the original order.
38	OrderQty	N	Not used by TXSE. FIX 5.0 SP2 requires this field (via OrderQtyData component), but TXSE does not process this field since all cancel requests cancel the full remaining quantity of the order identified by OrigClOrdID.

3.4 OrderReplaceRequest

OrderCancelReplaceRequest modifying an existing order. Order may maintain or lose queue priority based on modified fields.

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. 'G' for Order Cancel/Replace Request.
60	TransactTime	Y	Time request was initiated. Required by FIX 5.0, but not used by the Exchange.
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
41	OrigClOrdID	Y	Original ClOrdID. Must reference an existing order.
55	Symbol	Y	Security identifier root represented in CMS format. This field must not change from the value sent in the original order.
65	SymbolSfx	N	Security identifier suffix represented in CMS format. This field must not change from the value sent in the original order. Defaults to blank if not provided.
54	Side	Y	Enum Side . Side can only change between Sell and Sell-Short. Otherwise this field must match the value sent in the original order.
40	OrdType	N	Enum OrdType . Not used by TXSE. FIX 5.0 SP2 requires this field, but TXSE does not process this field since the order type cannot be modified via replace.
114	LocateReqd	N	Processed for Sell Short and Sell Short Exempt in which client affirms ability to borrow (isLocateRequired = false) or client does not affirm ability to borrow (isLocateRequired = true)
9000	LocateBroker	N	Used for short sale orders to identify the broker that has loaned the stock to settle the short sale. MPID should contain upper-case alpha only.
44	Price	N	Limit price. Must be greater than 0. Maximum price is \$100,000,000.00. Changing the price will cause the order to lose its priority in the book.
38	OrderQty	N	Number of shares. Must be greater than 0. Increase in the size of the order will cause the order to lose priority in the book.
18	ExecInst	N	Enum ExecInst .

Tag	Name	Req'd	Notes
9005	CancelAtEntryIfCrossed	N	True to cancel the order at entry if market conditions are crossed.
8001	SelfMatchScope	N	Enum SelfMatchScope . Determines the scope for self-match prevention. For additional information, refer to Self Match Prevention documentation .
2964	SelfMatchPreventionInstruction	N	Enum SelfMatchPreventionInstruction . Determines the behavior when orders from the same entity would match. For additional information, refer to Self Match Prevention documentation .
8000	PriceSlideInstruction	N	Enum PriceSlideInstruction . For additional information, refer to Display Price Slide Instructions documentation .
211	PegOffsetValue	N	The target relative to the reference base price in basis points of the NBBO spread.
1138	DisplayQty	N	Quantity to be displayed at one time with remaining quantity not displayed on the book.

4 FIX Application Messages from Exchange to Member

Messages sent from the Exchange to the Member.

4.1 ExecutionReportNew

ExecutionReport acknowledging a NewOrderSingle message. The order was successfully accepted (ExecType=NEW, OrdStatus=NEW).

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. '8' for Execution Report.
60	TransactTime	Y	The time this event was processed by the Exchange.
150	ExecType	Y	Enum ExecType . Execution type (NEW).
39	OrdStatus	Y	Enum OrdStatus . Order status (NEW).
40	OrdType	Y	Enum OrdType .
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
37	OrderID	Y	Exchange-assigned unique order identifier.
17	ExecID	Y	Exchange-assigned unique execution identifier.
55	Symbol	Y	Security identifier root represented in CMS format.
65	SymbolSfx	Y	Security identifier suffix represented in CMS format.
54	Side	Y	Enum Side . Determines direction of a trading action, including short-sell indicators.
114	LocateReqd	Y	True if a locate is required for short-sale, false otherwise.
9000	LocateBroker	N	Used for short sale orders to identify the broker that has loaned the stock to settle the short sale. MPID should contain upper-case alpha only.
44	Price	N	Limit price. Conditional on OrdType.
38	OrderQty	Y	Number of shares.
59	TimelnForce	Y	Enum TimelnForce .
18	ExecInst	N	Enum ExecInst .
9005	CancelAtEntryIfCrossed	Y	True to cancel the order at entry if market conditions are crossed.
528	OrderCapacity	Y	Enum OrderCapacity . Broker capacity in sending order.
109	ClientID	Y	Market Participant Identifier. MPID should contain upper-case alpha only.
9004	MemberGroup	Y	Member supplied grouping. Used for both self-match prevention and mass cancel, in conjunction with MPID or SenderComp.
8001	SelfMatchScope	Y	Enum SelfMatchScope . Determines the scope for self-match prevention. For additional information, refer to Self Match Prevention documentation .
2964	SelfMatchPreventionInstruction	Y	Enum SelfMatchPreventionInstruction . Determines the behavior when orders from the same entity would match. For additional information, refer to Self Match Prevention documentation .
8000	PriceSlideInstruction	Y	Enum PriceSlideInstruction . For additional information, refer to Display Price Slide Instructions documentation .
211	PegOffsetValue	N	The target relative to the reference base price in basis points of the NBBO spread.
110	MinQty	N	Number of shares that must be executed to execute any shares.

Tag	Name	Req'd	Notes
1138	DisplayQty	N	Quantity to be displayed at one time with remaining quantity not displayed on the book.
1087	DisplayMinIncr	N	Defines the maximum range to be used when calculating a random refresh quantity.
9001	MaxReplenishTimeRange	N	Defines the maximum range to be used when calculating a random refresh time, in nanoseconds.
126	ExpireTime	N	Timestamp at which the order's open quantity will be canceled, valid only with GTT time in force. For additional information, refer to GTT documentation .
9002	UserData	N	Pass-through field for use by clients, not used by the exchange.
151	LeavesQty	Y	Number of shares that were still available to execute after the execution. If this value is 0, then the order has been fully executed.
14	CumQty	Y	Cumulative quantity filled.
9008	RankPrice	N	Optional rank price, when different from limit price.
2828	CurrentDisplayPrice	N	Optional display price, when different from limit price.
115	OnBehalfOfCompID	N	Application name of the originating port. Only present in the Drop Copy protocol.

4.2 ExecutionReportRejected

ExecutionReport rejecting a NewOrderSingle message. The order was rejected (ExecType=REJECTED, OrdStatus=REJECTED).

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. '8' for Execution Report.
60	TransactTime	Y	The time this event was processed by the Exchange.
150	ExecType	Y	Enum ExecType . Execution type (REJECTED).
39	OrdStatus	Y	Enum OrdStatus . Order status (REJECTED).
40	OrdType	Y	Enum OrdType .
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
17	ExecID	Y	Exchange-assigned unique execution identifier.
55	Symbol	Y	Security identifier root represented in CMS format.
65	SymbolSfx	Y	Security identifier suffix represented in CMS format.
54	Side	Y	Enum Side . Determines direction of a trading action, including short-sell indicators.
114	LocateReqd	Y	True if a locate is required for short-sale, false otherwise.
9000	LocateBroker	N	Used for short sale orders to identify the broker that has loaned the stock to settle the short sale. MPID should contain upper-case alpha only.
44	Price	N	Limit price. Conditional on OrdType.
38	OrderQty	Y	Number of shares.
59	TimelnForce	Y	Enum TimelnForce .
18	ExecInst	N	Enum ExecInst .
9005	CancelAtEntryIfCrossed	Y	True to cancel the order at entry if market conditions are crossed.
528	OrderCapacity	Y	Enum OrderCapacity . Broker capacity in sending order.
109	ClientID	N	Market Participant Identifier. MPID should contain upper-case alpha only.

Tag	Name	Req'd	Notes
9004	MemberGroup	N	Member supplied grouping. Used for both self-match prevention and mass cancel, in conjunction with MPID or SenderComp.
8001	SelfMatchScope	N	Enum SelfMatchScope . Determines the scope for self-match prevention. For additional information, refer to Self Match Prevention documentation .
2964	SelfMatchPreventionInstruction	N	Enum SelfMatchPreventionInstruction . Determines the behavior when orders from the same entity would match. For additional information, refer to Self Match Prevention documentation .
8000	PriceSlideInstruction	N	Enum PriceSlideInstruction . For additional information, refer to Display Price Slide Instructions documentation .
211	PegOffsetValue	N	The target relative to the reference base price in basis points of the NBBO spread.
110	MinQty	N	Number of shares that must be executed to execute any shares.
1138	DisplayQty	N	Quantity to be displayed at one time with remaining quantity not displayed on the book.
1087	DisplayMinIncr	N	Defines the maximum range to be used when calculating a random refresh quantity.
9001	MaxReplenishTimeRange	N	Defines the maximum range to be used when calculating a random refresh time, in nanoseconds.
126	ExpireTime	N	Timestamp at which the order's open quantity will be canceled, valid only with GTT time in force. For additional information, refer to GTT documentation .
9002	UserData	N	Pass-through field for use by clients, not used by the exchange.
151	LeavesQty	Y	Number of shares that were still available to execute after the execution. If this value is 0, then the order has been fully executed.
14	CumQty	Y	Cumulative quantity filled.
103	OrdRejReason	Y	Enum OrdRejReason . Indicates the reason that a customer action was rejected by the exchange.
115	OnBehalfOfCompID	N	Application name of the originating port. Only present in the Drop Copy protocol.

4.3 ExecutionReportCanceled

ExecutionReport for an order cancellation, either solicited or unsolicited (ExecType=CANCELED, OrdStatus=CANCELED).

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. '8' for Execution Report.
60	TransactTime	Y	The time this event was processed by the Exchange.
150	ExecType	Y	Enum ExecType . Execution type (CANCELED).
39	OrdStatus	Y	Enum OrdStatus . Order status (CANCELED).
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
41	OrigClOrdID	Y	Original ClOrdID.
37	OrderID	Y	Exchange-assigned unique order identifier.
17	ExecID	Y	Exchange-assigned unique execution identifier.
55	Symbol	Y	Security identifier root represented in CMS format.

Tag	Name	Req'd	Notes
65	SymbolSfx	Y	Security identifier suffix represented in CMS format.
54	Side	Y	Enum Side . Determines direction of a trading action, including short-sell indicators.
151	LeavesQty	Y	Number of shares that were still available to execute after the execution. If this value is 0, then the order has been fully executed.
14	CumQty	Y	Cumulative quantity filled.
8003	CancelReason	Y	Enum CancelReason . Reason that an order was canceled.
115	OnBehalfOfCompID	N	Application name of the originating port. Only present in the Drop Copy protocol.

4.4 OrderCancelReject

OrderCancelReject rejecting an OrderCancelRequest that cannot be honored (CxlRejResponseTo=ORDER_CANCEL_REQUEST).

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. '9' for Order Cancel Reject.
60	TransactTime	Y	The time this event was processed by the Exchange.
434	CxlRejResponseTo	Y	Enum CxlRejResponseTo . Request being rejected (ORDER_CANCEL_REQUEST).
39	OrdStatus	Y	Enum OrdStatus . Order status (REJECTED, CANCELED, NEW, PARTIALLY_FILLED or FILLED).
11	CIOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
41	OrigCIOrdID	Y	Original CIOrdID.
37	OrderID	N	Exchange-assigned unique order identifier. Only present if the order was found. Absent if the order was not found (e.g., invalid OrigCIOrdID).
102	CxlRejReason	Y	Enum CxlRejReason .
115	OnBehalfOfCompID	N	Application name of the originating port. Only present in the Drop Copy protocol.

4.5 ExecutionReportReplaced

ExecutionReport confirming an OrderReplaceRequest was successfully applied (ExecType=REPLACED).

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. '8' for Execution Report.
60	TransactTime	Y	The time this event was processed by the Exchange.
150	ExecType	Y	Enum ExecType . Execution type (REPLACED).
39	OrdStatus	Y	Enum OrdStatus . Order status (NEW, PARTIALLY_FILLED, or FILLED).
11	CIOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
41	OrigCIOrdID	Y	Original CIOrdID.
37	OrderID	Y	Exchange-assigned unique order identifier.
17	ExecID	Y	Exchange-assigned unique execution identifier.
55	Symbol	Y	Security identifier root represented in CMS format.
65	SymbolSfx	Y	Security identifier suffix represented in CMS format.
54	Side	Y	Enum Side . Determines direction of a trading action, including short-sell indicators.
114	LocateReqd	Y	True if a locate is required for short-sale, false otherwise.

Tag	Name	Req'd	Notes
9000	LocateBroker	N	Used for short sale orders to identify the broker that has loaned the stock to settle the short sale. MPID should contain upper-case alpha only.
44	Price	N	Limit price.
38	OrderQty	Y	Number of shares.
18	ExecInst	N	Enum ExecInst .
9005	CancelAtEntryIfCrossed	Y	True to cancel the order at entry if market conditions are crossed.
8001	SelfMatchScope	N	Enum SelfMatchScope . Determines the scope for self-match prevention. For additional information, refer to Self Match Prevention documentation .
2964	SelfMatchPreventionInstruction	N	Enum SelfMatchPreventionInstruction . Determines the behavior when orders from the same entity would match. For additional information, refer to Self Match Prevention documentation .
8000	PriceSlideInstruction	N	Enum PriceSlideInstruction . For additional information, refer to Display Price Slide Instructions documentation .
211	PegOffsetValue	N	The target relative to the reference base price in basis points of the NBBO spread.
1138	DisplayQty	N	Quantity to be displayed at one time with remaining quantity not displayed on the book.
151	LeavesQty	Y	Number of shares that were still available to execute after the execution. If this value is 0, then the order has been fully executed.
14	CumQty	Y	Cumulative quantity filled.
9008	RankPrice	N	Optional rank price, when different from limit price.
2828	CurrentDisplayPrice	N	Optional display price, when different from limit price.
115	OnBehalfOfCompID	N	Application name of the originating port. Only present in the Drop Copy protocol.

4.6 OrderReplaceReject

OrderCancelReject rejecting an OrderReplaceRequest that cannot be honored (CxlRejResponseTo=ORDER_CANCEL_REPLACE_REQUEST).

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType . '9' for Order Cancel Reject.
60	TransactTime	Y	The time this event was processed by the Exchange.
434	CxlRejResponseTo	Y	Enum CxlRejResponseTo . Request being rejected (ORDER_CANCEL_REPLACE_REQUEST).
39	OrdStatus	Y	Enum OrdStatus . Order status (REJECTED, CANCELED, NEW, PARTIALLY_FILLED or FILLED).
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
41	OrigClOrdID	Y	Original ClOrdID.
37	OrderID	N	Exchange-assigned unique order identifier. Only present if the order was found. Absent if the order was not found (e.g., invalid OrigClOrdID).
54	Side	Y	Enum Side . Determines direction of a trading action, including short-sell indicators.
114	LocateReqd	Y	True if a locate is required for short-sale, false otherwise.

Tag	Name	Req'd	Notes
9000	LocateBroker	N	Used for short sale orders to identify the broker that has loaned the stock to settle the short sale. MPID should contain upper-case alpha only.
44	Price	N	Limit price.
38	OrderQty	N	Number of shares.
18	ExecInst	N	Enum ExecInst .
9005	CancelAtEntryIfCrossed	Y	True to cancel the order at entry if market conditions are crossed.
8001	SelfMatchScope	N	Enum SelfMatchScope . Determines the scope for self-match prevention. For additional information, refer to Self Match Prevention documentation .
2964	SelfMatchPreventionInstruction	N	Enum SelfMatchPreventionInstruction . Determines the behavior when orders from the same entity would match. For additional information, refer to Self Match Prevention documentation .
8000	PriceSlideInstruction	N	Enum PriceSlideInstruction . For additional information, refer to Display Price Slide Instructions documentation .
211	PegOffsetValue	N	The target relative to the reference base price in basis points of the NBBO spread.
1138	DisplayQty	N	Quantity to be displayed at one time with remaining quantity not displayed on the book.
102	CxlRejReason	Y	Enum CxlRejReason .
115	OnBehalfOfCompID	N	Application name of the originating port. Only present in the Drop Copy protocol.

4.7 ExecutionReportTrade

ExecutionReport for an order execution (ExecType=TRADE, OrdStatus=PARTIALLY_FILLED or FILLED).

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. '8' for Execution Report.
60	TransactTime	Y	The time this event was processed by the Exchange.
150	ExecType	Y	Enum ExecType . Execution type (TRADE).
39	OrdStatus	Y	Enum OrdStatus . Order status (PARTIALLY_FILLED or FILLED).
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
37	OrderID	Y	Exchange-assigned unique order identifier.
17	ExecID	Y	Exchange-assigned unique execution identifier.
55	Symbol	Y	Security identifier root represented in CMS format.
65	SymbolSfx	Y	Security identifier suffix represented in CMS format.
54	Side	Y	Enum Side . Determines direction of a trading action, including short-sell indicators.
151	LeavesQty	Y	Number of shares that were still available to execute after the execution. If this value is 0, then the order has been fully executed.
14	CumQty	Y	Cumulative quantity filled.
31	LastPx	Y	The price at which these shares were executed.
32	LastQty	Y	Number of shares that were executed.
851	LastLiquidityInd	Y	Enum LastLiquidityInd .
9730	TradeLiquidityIndicator	Y	Enum TradeLiquidityIndicator .
115	OnBehalfOfCompID	N	Application name of the originating port. Only present in the Drop Copy protocol.

4.8 ExecutionReportReplenished

ExecutionReport indicating a reserve order's display quantity has been replenished (ExecType=RESTATED).

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. '8' for Execution Report.
60	TransactTime	Y	The time this event was processed by the Exchange.
150	ExecType	Y	Enum ExecType . Execution type (RESTATED).
39	OrdStatus	Y	Enum OrdStatus . Order status (NEW or PARTIALLY_FILLED).
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
37	OrderID	Y	Exchange-assigned unique order identifier.
17	ExecID	Y	Exchange-assigned unique execution identifier.
55	Symbol	Y	Security identifier root represented in CMS format.
65	SymbolSfx	Y	Security identifier suffix represented in CMS format.
54	Side	Y	Enum Side . Determines direction of a trading action, including short-sell indicators.
44	Price	N	Limit price.
38	OrderQty	Y	Number of shares.
151	LeavesQty	Y	Number of shares that were still available to execute after the execution. If this value is 0, then the order has been fully executed.
14	CumQty	Y	Cumulative quantity filled.
378	ExecRestatementReason	Y	Enum ExecRestatementReason . Reason for order restatement events triggered by the Exchange.
1138	DisplayQty	Y	Quantity to be displayed at one time with remaining quantity not displayed on the book.
115	OnBehalfOfCompID	N	Application name of the originating port. Only present in the Drop Copy protocol.

4.9 ExecutionReportRepriced

ExecutionReport indicating a pegged order's price has been adjusted due to a reference price change (ExecType=RESTATED).

Tag	Name	Req'd	Notes
35	MsgType	Y	Enum MsgType. '8' for Execution Report.
60	TransactTime	Y	The time this event was processed by the Exchange.
150	ExecType	Y	Enum ExecType . Execution type (RESTATED).
39	OrdStatus	Y	Enum OrdStatus . Order status (NEW or PARTIALLY_FILLED).
11	ClOrdID	Y	Client-specified order identifier with maximum length of 20 characters.
37	OrderID	Y	Exchange-assigned unique order identifier.
17	ExecID	Y	Exchange-assigned unique execution identifier.
55	Symbol	Y	Security identifier root represented in CMS format.
65	SymbolSfx	Y	Security identifier suffix represented in CMS format.
54	Side	Y	Enum Side . Determines direction of a trading action, including short-sell indicators.
44	Price	N	Limit price.
38	OrderQty	Y	Number of shares.
151	LeavesQty	Y	Number of shares that were still available to execute after the execution. If this value is 0, then the order has been fully executed.
14	CumQty	Y	Cumulative quantity filled.

Tag	Name	Req'd	Notes
378	ExecRestatementReason	Y	Enum ExecRestatementReason . Reason for order restatement events triggered by the Exchange.
9008	RankPrice	Y	The rank price. Only present on re-priced orders.
2828	CurrentDisplayPrice	N	The display price, when different from rank price. Only present on re-priced orders.
115	OnBehalfOfCompID	N	Application name of the originating port. Only present in the Drop Copy protocol.