MEMO SBE

V1.8 Rev-C

11/17/22

This specification is being provided to you strictly for informational purposes solely for the purpose of developing or operating systems that interact with the Members Exchange, or MEMX. All proprietary rights and interest in this specification and the information contained herein shall be vested in MEMX and all other rights including, but without limitation, patent, registered design, copyright, trademark, service mark, connected with this publication shall also be vested in MEMX. 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 MEMX. MEMX reserves the right to withdraw, modify, or replace the specification at any time, without notice. No obligation is made by MEMX regarding the level, scope, or timing of MEMX's implementation of the functions or features discussed in this specification.

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

Table of Contents

< THIS PAGE INTENTIONALLY LEFT BLANK>

1 Overview

MEMO is a binary protocol used by members to submit orders to the MEMX exchange.

2 Architecture

A MEMO session is a sequenced messaging protocol that uses fixed width binary messages.

The MEMO wire format uses an SBE-encoded subset of FIX 5.0 SP2 tags and messages.
More information about the FIX protocol and associated tag numbers can be found on the Financial Information Exchange (FIX) Protocol version 5.0 Service Pack 2 page.

MEMO messages are transported using the MEMX-TCP protocol to provide reliable ordered transmission of messages between clients and servers. Connected clients should always use the streaming mode of MEMX-TCP.

Making MEMX-TCP requests for message replay will result in the client being disconnected, as per the MEMX-TCP specification. Heartbeat and heartbeat timeout intervals for a MEMO connection may be configured via the MEMO port support request process.

3 Encoding

The MEMO protocol uses the FIX Trading Community's Simple Binary Encoding (SBE) to specify message encoding. More information about SBE can be found at the FIX SBE XML Primer.

MEMO is always encoded in Big Endian byte order.

3.1 SBE Data Types

All encoding and decoding for SBE is centered around a set of basic primitive types.

For more information on primitive type encoding, see the SBE specification.

Type	Length (bytes)	Description	Value Range	Null Value	Null Value (Hex)
CHAR	1	ASCII Character	`32` to `126` (printable ASCII)	`0`	`0x00`
INT8	1	Signed Integer	`-127` to `127`	`-128`	`0x80`
INT16	2	Signed Integer	`-32767` to `32767`	`-32768`	`0x8000`
INT32	4	Signed Integer	`-2^31 + 1` to `2^31 - 1`	`-2^31`	`0x80000000`
INT64	8	Signed Integer	`-2^63 + 1` to `2^63 - 1`	`-2^63`	`0x8000000000000000`
UINT8	1	Unsigned Integer	`0` to `254`	`255`	`0xFF`
UINT16	2	Unsigned Integer	`0` to `65534`	`65535`	`0xFFFF`
UINT32	4	Unsigned Integer	`0` to `2^32-2`	`2^32 - 1`	`0xFFFFFFFF`
UINT64	8	Unsigned Integer	`0` to `2^64 - 2`	`2^64 - 1`	`0xFFFFFFFFFFFFFFFF`

The MEMO specification does not use the SBE floating point data types.

3.1.1 Price

Prices are encoded as a fixed-point scaled decimal, consisting of a signed long (INT64) mantissa and a constant exponent of -6.

Type Name	Length	Type	Description
Mantissa	8	INT64	The fixed-point decimal representation of the price
Exponent	N/A	INT8	MEMO uses a constant exponent of -6 for all prices.

For example, a mantissa of 1234567 (0x12D687) with the constant exponent of -6 represents the decimal number 1.234567 and would appear encoded on the wire as the hex value `000000000012D687`.

3.1.2 UTCTimestampNanos

Fields with the UTCTimestampNanos type represent a timestamp in Coordinated Universal Time (UTC) which began on the UNIX epoch (January 1, 1970 at 00:00:00 UTC). UTCTimestampNanos has the time unit of nanoseconds and is encoded as follows:

Type Name	Length	Type	Description
Time	8	UINT64	UTC timestamp since unix epoch with nanosecond precision.
Unit	N/A	UINT8	Unit of time. UTCTimestampNanos are represented in nanoseconds. This field is constant (value=9) and as such will not be transferred on the wire.

3.2 SBE Header

SBE includes a header for each message. The SBE header is followed by the SBE body for the message.

The SBE message header contains the fields that allows the decoder to identify what codec should be used for a message.

1. **blockLength**: The length of the message. Note: this does not include the header length.
2. **templateID**: The identifier for the template type of the message that is to follow.
3. **schemaID**: The identifier for the schema the message belongs to.
4. **version**: The version of the schema allowing for extension. MEMX packs two pieces of information into the (UINT16) version field, a major version and a minor update version. So, for example a version of 258 = 0x0102 indicates major version 1, minor update 2.

The MEMO SBE header appears on the wire as:

Field	Offset	Length	Type	Description
BlockLength	0	2	UINT16	The number of bytes in the message body (does not include the header bytes). Note that MEMO messages do not use repeating groups or variable-length fields.
TemplateID	2	1	UINT8	Identifier of the message template (ie. The message type)
SchemaID	3	1	UINT8	The identifier of a message schema. SchemaID=1 for MEMO version 1
Version	4	2	UINT16	The version number of the message schema that was used to encode a message. MEMX packs two pieces of information into the (UINT16) version field, a major version and a minor update version. For example, a version of 258 = 0x0102 indicates major version 1, minor update 2.

4 Message Field Types

All messages are composed of fields. Each field has a type. This section defines the non-primitive field types, their underlying type on the wire, acceptable values, and their descriptions.

4.1 CancelReasonCode (MEMX custom FIX tag 21004)

CancelReasonCode represents the reason the order was canceled by the system.

CancelReasonCode is a 1-byte UINT8 value.

Value	Name	Description
0	Other	This order was canceled for some other reason not listed.
1	UserRequestedCancel	The client sent a OrderCancelRequest or OrderMassCancelRequest for this order.
4	EndOfTrading	The order was sent with the DAY time in force set, and the DAY trading session completed.
5	LimitUpLimitDown	The price of the order fell outside market LULD bands, and the re-pricing modifier was not specified on the order.
6	Halted	The market on the order's security was halted.
7	ExchangeSupervisory	Operational or supervisory actions taken by MEMX resulted in the cancellation of this order.
8	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.
9	LockOrCrossBook	The order was not externally routable, and market conditions would have resulted in this order crossing or locking the order book.
10	SelfTradePrevention	This or another associated order's specified self trade prevention behavior triggered the cancellation of this order.
11	InsufficientQuotes	The order was cancelled because there are insufficient quotes on the book for the symbol.
12	NonCompliantPrice	The order was cancelled because the price in the order was non-compliant.
13	ParticipantDisconnect	The participant directed that their orders should be canceled when the trading system detects a disconnection, and the participant disconnected.
14	OrderNotBookable	Order is not of bookable type (this may include market orders, IOC, FOK, etc)
15	TradeProtectionLimits	The price of the order fell outside market trade protection limits rule, and the re-pricing modifier was not specified on the order.
16	UnableToRoute	Order was canceled because it was externally routable but could not be routed.
17	FirmDisabled	Order was canceled because the firm was disabled
18	MPIDDisabled	Order was canceled because the MPID was disabled
19	AccountDisabled	Order was canceled because the Account was disabled
20	NotionalExposure RiskBreached	Order was canceled because a Notional Exposure Risk Rule was breached

4.2 StpGroupIDType (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 a 2-byte UINT16 value.

Value	Name	Description
0	Firm Scope	Prevents orders from the same firm trading with each other
1	MPID Scope	Prevents orders from the same MPID trading with each other
2	Account Scope	Prevents orders from the same account trading with each other
3-65534	Custom Scope	Prevents orders with matching StpGroupID across the firm from trading with each other
65535	NULL	The UINT16 Null value (0xFFFF) disables custom self trade prevention grouping.

4.3 CancelRejectReasonCode (FIX tag 102)

Identifies the reason why an order cancellation request was rejected.

CancelRejectReasonCode is a 1-byte UINT8 value.

Value	Name	Description
0	Other	This order was rejected for some other reason not listed.
1	MissingSymbol	The order is rejected because the Symbol/SymbolSfx was not specified.
2	MissingClOrdId	The order is rejected because the ClOrdID was not specified.
3	MissingOrigOrderIdentifiers	The order is rejected because the OrigOrderID was not specified.
4	AmbiguousOrigOrderIdentifiers	The order is rejected because the OrigOrderID was ambiguous, does not uniquely identify an order in the system.
5	UnknownOrigOrder	An order with the specified OrigOrderID was not found.
6	OrigOrderSymbolNotMatching RequestSymbol	The order is rejected because the Symbol/SymbolSfx does not match the Symbol/SymbolSfx on the original order.
7	MissingLocate	The order request is malformed, missing the LocateReqd field.
8	InvalidOrderQuantity	The supplied order’s quantity was invalid.
9	InvalidSymbol	Request symbol is not in MEMX symbols list.
10	InvalidLimitPrice	Request Limit Price is invalid.
11	InvalidLimitPriceIncrement	Request Limit Price increment is invalid.

Value	Name	Description
12	InvalidLocate	The order request is malformed. Request must contain LocateReqd = 'N' field for short sales.
13	SymbolHaltedOrPaused	The order was rejected because the symbol is halted or paused.
14	ExchangeClosed	The order was rejected because the exchange is closed.
15	DuplicateClOrdID	The supplied ClOrderID duplicated another order for this account.
16	OrderSizeExceedsLimit	The supplied order's quantity exceeds exchange-level order size limits.
17	ExceededMaxNotionalOrderAmt	This order violated one or more notional risk checks.
18	UnsupportedDisplayQuantityChange	Change of DisplyQty in replace request not supported.
19	UnsupportedOrdTypeChange	Change of OrdType in replace request not supported.
20	UnsupportedSideChange	Change of side in replace request not supported.
21	UnsupportedQuantityChange	Change of quantity in replace request not supported.
22	OrderInPendingState	There is a pending request on this order.
23	BlockSessionRiskRuleViolated	Session Risk Rule was violated
24	BlockSellShortRiskRuleViolated	Sell Short Risk Rule was violated
25	MaxSharesPerOrderRiskRuleBreach	Max Shares Per Order Risk Rule was violated
26	NoNBBOAvailable	No NBBO available
27	MaxNotionalValuePerOrderRiskRuleBreach	Max Notional Value Per Order Risk Rule was violated
28	MaxADVPercentPerOrderRiskRuleBreach	Max ADV Percent Per Order Risk Rule was violated
29	PricePercentCollarRiskRuleViolated	Price Percent Collar Risk Rule was violated
30	PriceValueCollarRiskRuleViolated	Price Value Collar Risk Rule was violated
31	HardToBorrowSecurityRiskRuleViolated	HardToBorrowSecurityRiskRuleViolated
32	InvalidSide	Missing or invalid side for replace request.
33	InvalidOrdType	Missing or invalid OrdType for replace request.
34	InvalidClOrdId	The ClOrdID is invalid and cannot be used.

4.4 CxlRejResponseToType (FIX tag 434)

CxlRejResponseToType identifies the request type that a cancel reject was issued for.

CxlRejResponseToType is a 1-byte UINT8 value.

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

4.5 MassCancelRejectReasonCode

Identifies the reason why a mass cancellation request was rejected.

MassCancelRejectReasonCode is a 1-byte UINT8 value.

Value	Name	Description
0	Other	This order was rejected for some other reason not listed.
1	UnknownProduct	Unknown requested Symbol + Suffix combination
2	UnknownSide	Unknown requested Side
3	UnknownGroupId	Unknown request GroupId for firm
4	HigherPriceLowerOr EqualToLowerPrice	When both HigherThanPrice and LowerThanPrice fields are configured on MassCancelRequest HigherThanPrice must be greater than LowerThanPrice.
5	ProductMissingFor PriceRestriction	Cancel per price restriction (HigherPrice / LowerPrice) requires presence of valid product on the request.
6	DuplicateClOrdID	The ClOrdID is a duplicate and cannot be reused.
7	MalformedRequest MissingClOrdIdField	The request is malformed, it is missing the ClOrdId field.
8	InvalidCancelGroupId	The cancel group ID supplied is invalid for the account.
9	InvalidClOrdId	The ClOrdID is invalid and cannot be used.
10	InvalidLowerPrice	The LowerPrice specified was invalid.
11	InvalidHigherPrice	The HigherPrice specified was invalid.

4.6 DispMethodType (FIX tag 1084)

DispMethodType indicates the method used when refreshing displayed quantity. Undisclosed should be specified for Pegged and Market orders. This field should be set to the null value (0xFF) for a fully displayed Limit order. Currently if the null value (0xFF) is specified for a Pegged or Market order it will be interpreted as Undisclosed but this is not recommended as this behavior may change in a future revision of the specification.

DispMethodType is a 1-byte UINT8 value.

Value	Name	Description
1	Initial	Use the DisplayQty specified on the order.
2	Random	Use the random algorithm to refresh the displayed quantity after a fill.
3	Undisclosed	This order will never be displayed. The DisplayQty field is ignored.

4.7 ReserveReplenishTimingType (MEMX custom FIX tag 21006)

Defines the replenishment timing behavior for a reserve order.

ReserveReplenishTimingType is a 1-byte UINT8 value.

Value	Name	Description
1	Immediate	Update the reserve order immediately after a fill.
2	Random	Update the reserve order a random time interval after a fill.

4.8 ExchangeCode (FIX tag 30)

ExchangeCode indicates the market of execution for last fill, or an indication of the market where an order was routed.

ExchangeCode is a 1-byte CHAR value.

Value	Name
A	NYSE American
B	Nasdaq BX
C	NYSE National
H	MIAX Pearl
J	CBOE EDGE-A
K	CBOE EDGE-X
L	LTSE
M	NYSE Chicago
N	NYSE
P	NYSE ARCA
Q	Nasdaq
U	MEMX
V	IEX
X	Nasdaq PSX
Y	CBOE BATS-Y
Z	CBOE BATS-Z

4.9 ExecInstType Bitset (FIX tag 18)

The ExecInstType provides instructions for order handling on exchange.

ExecInstType is a 2-byte UINT16 value.

Type Name	Length	Type	Values	Description
ExecInstType	2	UINT16	Bit 0 – ParticipateDoNotInitiate (Post only) Bit 1 – IntermarketSweep (ISO) Bit 2 – ExternalRoutingNotAllowed (Book Only)	Enable an execution instruction by setting a bit. Disable an execution instruction by clearing the bit. Bit 0 is the least significant bit.

4.10 ExecRestatementType (FIX tag 378)

Defines the reason for a restatement.

ExecRestatementType is a 1-byte UINT8 value.

Value	Name	Description
1	OrderReprice	The order's booking price was updated.
2	SelfTradePrevention	The order quantity was reduced to prevent a wash trade.

4.11 LastLiquidityIndType (FIX tag 851)

Indicates the liquidity code for this execution.

LastLiquidityIndType is a 1-byte UINT8 value.

Value	Name	Description
1	AddDisplayed	This fill added displayed liquidity to the MEMX book.
2	Removed	This fill removed liquidity from the MEMX book.
3	Routed	This fill was routed to another market, where it removed liquidity.
4	Cross	This value is currently not used.
5	AddHidden*	This fill added hidden liquidity to the MEMX book. MEMX-defined value – Not present in FIX tag 851
6	AddMidpointPeg*	This fill added liquidity to the MEMX book via a midpoint peg order. MEMX-defined value – Not present in FIX tag 851
7	AddDisplayedNbboImprove*	This fill added displayed liquidity to the MEMX book which improved the NBBO. MEMX-defined value – Not present in FIX tag 851
8	AddNbboJoin*	The order added displayed liquidity that caused the MEMX quote to join the existing NBBO MEMX-defined value – Not present in FIX tag 851
10	ImmediateMidpointRemoveOnEntry*	A non-bookable midpoint order removed liquidity at the midpoint price on entry – Not present in FIX tag 851
11	AddDisplayedPriceImprovement*	A booked displayed order received price improvement on their working price – Not present in FIX tag 851
12	AddHiddenPriceImprovement*	A booked hidden order received price improvement on their working price – Not present in FIX tag 851
101	RetailAddDisplayed*	This fill added displayed liquidity to the MEMX book. MEMX-defined value – Not present in FIX tag 851
102	RetailRemoved*	This fill removed liquidity from the MEMX book. MEMX-defined value – Not present in FIX tag 851
103	RetailRouted*	This fill was routed to another market, where it removed liquidity. MEMX-defined value – Not present in FIX tag 851
104	RetailCross*	This value is currently not used. MEMX-defined value – Not present in FIX tag 851
105	RetailAddHidden*	This fill added hidden liquidity to the MEMX book. MEMX-defined value – Not present in FIX tag 851
106	RetailAddMidpointPeg*	This fill added liquidity to the MEMX book via a midpoint peg order. MEMX-defined value – Not present in FIX tag 851
107	RetailAddDisplayedNbboImprove*	This fill added displayed liquidity to the MEMX book which improved the NBBO. MEMX-defined value – Not present in FIX tag 851

Value	Name	Description
108	RetailAddNbboJoin*	The order added displayed liquidity that caused the MEMX quote to join the existing NBBO. MEMX-defined value – Not present in FIX tag 851
109	RetailRemovedOnEntry*	Retail order removed liquidity on entry – Not present in FIX tag 851
110	RetailImmediateMidpointRemoveOnEntry*	A non-bookable retail midpoint order removed liquidity at the midpoint price on entry – Not present in FIX tag 851
111	RetailAddDisplayedPriceImprovement*	A booked displayed retail order received price improvement on their working price – Not present in FIX tag 851
112	RetailAddHiddenPriceImprovement*	A booked hidden retail order received price improvement on their working price – Not present in FIX tag 851

The '*' – Indicates this is a MEMX value, it is not present in the standard values for FIX tag 851.

4.12 OrdStatusType (FIX tag 39)

Indicates the status of an order.

OrdStatusType is a 1-byte UINT8 value.

Value	Name
1	New
2	PartialFilled
3	Filled
4	Canceled
5	PendingCancel
6	Rejected
7	PendingNew
8	PendingReplace
9	Expired

4.13 OrdType (FIX tag 40)

Indicates the type of an order.

OrdType is a 1-byte UINT8 value.

Value	Name
1	Market
2	Limit
3	Pegged

4.14 OrderCapacityType (FIX tag 528)

Designates the capacity of the firm placing the order.

OrderCapacityType is a 1-byte UINT8 value.

Value	Name
1	Agency
2	Principal
3	Riskless Principal

4.15 CustOrderCapacityType (FIX tag 582)

Capacity of customer placing the order.

CustOrderCapacityType is a 1-byte UINT8 value.

Value	Name
1	MemberTradingOnTheirOwnAccount
5	RetailCustomer

4.16 OrderRejectReasonCode (FIX tag 103)

Identifies the reason why a new order or an order-replace was rejected by the system.

OrderRejectReasonCode is a 1-byte UINT8 value.

Value	Name	Description
0	Other	This order was rejected for some other reason not listed.
1	MissingSymbol	The order did not supply the symbol.
2	MissingLocate	The order did not supply the locate.
3	MissingClOrdId	The order did not supply the ClOrdId.
4	MissingSide	The order did not supply the side.
5	MissingOrderQuantity	The order did not supply the quantity.
6	MissingOrderType	The order did not supply the order type.
7	MissingTimeInForce	The order did not supply the time in force.
8	MissingOrderCapacity	The order did not supply the order capacity.
9	MissingExecInst	The order did not supply the execution instructions
10	MissingLimitPrice	The order did not supply the limit price.

Value	Name	Description
11	MissingMaxFloor	The order did not supply the max floor.
12	MissingReserveReplenish AmountType	The order did not supply the reserve replenish amount.
13	MissingReserveReplenish TimeType	The order did not supply the reserve replenish time.
14	MissingRandomReplenish Value	The order did not supply the random replenish value.
15	MissingRepriceFrequencyType	The order did not supply the reprice frequency type.
16	MissingRepriceBehaviorType	The order did not supply the reprice behavior type.
17	MissingCustomerCapacityType	The order did not supply the customer capacity type.
18	MissingExpireTime	The order did not supply the expire time.
19	MissingPegType	The order did not supply the peg type.
20	InvalidModifierForOrderType	The order contained an invalid modifier for the order type.
21	InvalidModifiersCombination	The order contained a set of modifiers that do not work together.
22	InvalidTradingSession ForOrderType	The order contained an invalid trading session.
23	InvalidTimeInForce ForOrderType	The order contained an invalid time in force type.
24	InvalidMinQuantity	The order contained an invalid min quantity.
25	InvalidOrderQuantity	The order contained an invalid order quantity.
26	InvalidSide	The order contained an invalid side.
27	InvalidOrderType	The order contained an invalid order type.
28	InvalidTimeInForce	The order contained an invalid time in force.
29	InvalidOrderCapacity	The order contained an invalid capacity.
30	InvalidCustomerCapacity	The order contained an invalid customer capacity.
31	InvalidSymbol	The order contained an invalid symbol.
32	InvalidExpireTime	The order contained an invalid expire time.
33	InvalidLimitPrice	The order contained an invalid limit price.
34	InvalidLimitPriceIncrement	The order contained an invalid limit price increment.
35	InvalidMaxFloor	The order contained an invalid max floor.
36	InvalidRandomReplenish Value	The order contained an invalid random replenish value.
37	InvalidRandomReplenish ValueForReserveType	The order contained an invalid random replenish for reserve.
38	InvalidReserveReplenish AmountType	The order contained an invalid reserve replenish amount.

Value	Name	Description
39	InvalidReserveReplenish TimeType	The order contained an invalid reserve replenish time.
40	InvalidRepriceFrequencyType	The order contained an invalid reprice frequency type.
41	InvalidRepriceBehaviorType	The order contained an invalid reprice behavior type.
42	InvalidRepriceBehavior ForRepriceFrequency	The order contained an invalid reprice behavior for reprice frequency.
43	InvalidMPIDValue	The order contained an invalid MPID.
44	InvalidPegType	The order contained an invalid peg type.
45	InvalidModifierForPegType	The order contained an invalid modifier for peg type.
46	InvalidLocate	The order contained an invalid locate value.
47	SymbolHaltedOrPaused	The order was on a symbol that is currently halted or paused.
48	ExchangeClosed	The order arrived while the exchange was closed.
49	DuplicateClOrdID	The order used a ClOrdId that has been used before.
50	OrderSizeExceedsLimit	The order exceeded exchange size limit.
51	OrderNotionalExceedsLimit	The order's notional value exceeded exchange notional value limit.
52	BlockISORiskRuleViolated	The order violated the firm's ISO risk rule.
53	BlockSessionRiskRuleViolated	The order violated the firm's Session risk rule.
54	BlockSellShortRiskRuleViolated	The order violated the firm's short sale risk rule.
55	BlockNonTestSymbols RiskRuleViolated	The order violated the firm's test symbol trading risk rule.
56	MaxSharesPerOrder RiskRuleBreach	The order breached the firm's max shares per order risk rule.
57	MaxNotionalValuePerOrder RiskRuleBreach	The order breached the firm's max notional value per order risk rule.
58	PricePercentCollar RiskRuleViolated	The order violated the firm's price percent collar risk rule.
59	PriceValueCollar RiskRuleViolated	The order violated the firm's value collar risk rule.
60	MaxADVPercentPerOrder RiskRuleBreach	The order violated the firm's max ADV (average daily volume) percentage risk rule.
61	DailyGrossNotionalExposure RiskRuleBreach	The order breached the firm's daily gross notional exposure risk rule.
62	DailyNetNotionalExposure RiskRuleBreach	The order breached the firm's daily net notional exposure risk rule.
63	MaxNumDuplicateOrders RiskRuleBreach	The order breached the firm's max number of duplicate orders risk rule.
64	MaxOrderRateRiskRuleBreach	The order breached the firm's max number of orders per second risk rule.
65	RestrictedSecurity RiskRuleViolated	The order was for a symbol on the firm's restricted security list.

Value	Name	Description
66	HardToBorrowSecurity RiskRuleViolated	The order was for a symbol on the firm's hard to borrow list.
67	InvalidSelfTradePreventionConfiguration	The order was rejected for a self trade prevention configuration setting.
68	InvalidSelfTradePreventionType	The order was rejected for an invalid self trade prevention value.
69	InvalidRiskGroupId	The order was rejected for an invalid risk group id value.
70	FirmDisabled	The order was rejected because the firm sending the order has been disabled.
71	MPIDDisabled	The order was rejected because the MPID used in the order has been disabled.
72	AccountDisabled	The order was rejected because the account sending the order has been disabled.
73	NoNBBOAvailable	The order was rejected because there is no NBBO available.
74	CannotTradeNonTestSymbol	The order was rejected because the account is set in Test mode cannot trade non-test symbols.
75	MissingFirm	The order was rejected because the Exchange cannot find the firm associated with the order.
76	MissingAccount	The order was rejected because the Exchange cannot find the account associated with the order.
77	MissingMPID	The order was rejected because the Exchange cannot find the MPID associated with the order.
78	MissingRiskGroup	The order was rejected because the Exchange cannot find the risk group associated with the order.
79	DailyMarketOrderGrossNotionalExposureRiskRuleBreach	The order breached the firm's daily market order gross notional exposure risk rule.
80	DailyMarketOrderNetNotionalExposureRiskRuleBreach	The order breached the firm's daily market order net notional exposure risk rule.
81	MissingDispMethodType	The order did not supply the DispMethodType
82	MissingFirmRiskSetting	Risk settings are missing for the firm
83	InvalidAccountMPIDToFirm	Invalid configuration of account and MPID to firm
84	InvalidPegOffsetValue	Peg offset value is set to sub penny
85	InvalidDispMethodType	Display method type is invalid
86	MissingCancelGroupId	The order did not supply the cancel group ID
87	InvalidCancelGroupId	The order was rejected for an invalid cancel group ID value
88	MissingSTPGroupId	The order did not supply the STP group ID
89	InvalidSTPGroupId	The order was rejected for an invalid STP group ID value

Value	Name	Description
90	InvalidClOrdId	The order was rejected for an invalid ClOrdID.

4.17 PegType (FIX tag 1094)

Defines the type of peg used in a peg order.

PegType is a 1-byte UINT8 value.

Value	Name	Description
1	MidPricePeg	Order price is pegged at the midpoint of the National Best Bid and Offer (NBBO)
2	PrimaryPeg	Order price is pegged to follow the best bid, when buying a security, and the best offer, when selling a security

4.18 RepriceFrequencyType (MEMX custom FIX tag 21020)

Defines the type of reprice used when marking an order for regulatory compliance.

The type specified is applied to display-eligible orders for compliance with RegNMS and to both display-eligible and hidden orders for compliance with RegSHO and LULD.

Note: Hidden orders are not re-priced for compliance with RegNMS.

RepriceFrequencyType is a 1-byte UINT8 value.

Value	Name	Description
1	SingleReprice	An order will be repriced upon entry for compliance with RegNMS, RegSHO, and LULD.
2	ContinuousReprice	An order will continuously be repriced after being booked for compliance with RegNMS, RegSHO, and LULD.
3	None	No repricing will be done.

4.19 RepriceBehaviorType (MEMX custom FIX tag 21021)

Defines the market conditions for which to apply repricing for.

RepriceBehaviorType is a 1-byte UINT8 value.

Value	Name	Description
1	RepriceLockCancelCross	An order will only be repriced if the market is locked, cancel if market is crossed.
2	RepriceLockRepriceCross	An order will be repriced if the market is locked or crossed.

4.20 SelfTradePreventionType (MEMX 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 an opposite side resting interest with the same self trade prevention identifier, the behavior specified by this value will be triggered.

SelfTradePreventionType is a 1-byte UINT8 value.

Value	Name	Description
1	CancelNewest	If triggered, this order (the incoming order) will be canceled.
2	CancelOldest	If triggered, the resting order will be canceled.
3	Decrement AndCancel	If triggered and both orders have equivalent size, both will be canceled. If triggered and orders have differing sizes, the smaller order will be canceled, and the larger order will be decremented by the size of the smaller order and the balance will remain on the book.
4	CancelBoth	If triggered, both the incoming and the resting orders will be canceled.
5	Cancel Smallest	If triggered and both orders have equivalent size, both will be canceled. If triggered and orders have differing size, the smaller order will be canceled.

4.21 SideType (FIX tag 54)

Defines the side of an order.

SideType is a 1-byte UINT8 value.

Value	Name
1	Buy
2	Sell
3	SellShort
4	SellShortExempt

4.22 TimeInForceType (FIX tag 59)

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

TimeInForceType is a 1-byte UINT8 value.

Value	Name	Description
1	Day	The order is entered for execution during the pre-market and market periods.
2	Immediate Or Cancel (IOC)	The order shall be partially or completely executed immediately or canceled outright.
3	Fill or Kill (FOK)	The order shall be completely filled immediately or canceled.
4	GoodForTime	The order shall be entered for execution during the supplied time interval.
5	Regular Hours Only (RHO)	The order is entered for execution during the market period. MEMX-defined value – not present in FIX tag 59

5 Messages

This section defines the messages that make up the protocol.

5.1 Client To Server

5.1.1 NewOrderSingle

The new order message type is used by institutions wishing to electronically submit securities orders to the exchange for execution.

This message corresponds to FIX message type D

Field	Offset	Len	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 92 (0x5C), template = 1
ClOrdID	6	16	CHAR	11	Y	Unique identifier of the order as assigned by the client.
MPID	22	4	CHAR	21007	N	The MPID associated with the new order, if it is not supplied, the MPID assigned as the default MPID for the Port/Account is associated with the new order.
Symbol	26	6	CHAR	55	Y	CMS symbol tradable instrument.
SymbolSfx	32	6	CHAR	65	N	CMS symbol suffix. Additional information about the security (e.g. preferred, warrants, etc.).
Side	38	1	SideType	54	Y	Side of order.
OrderQty	39	4	UINT32	38	Y	Quantity ordered. This represents the number of shares. Number of shares may not exceed 1MM. The notional value of an order may not exceed $30MM.
OrdType	43	1	OrdType	40	Y	Type of the order.
Price	44	8	Price	44	N	Price per unit of quantity (e.g. per share).
TimeInForce	52	1	TimeInForceType	59	Y	Defines the time during which an order is eligible for execution.
OrderCapacity	53	1	OrderCapacity Type	528	Y	Designates the capacity of the firm placing the order.
CustOrderCapacity	54	1	CustOrder CapacityType	582	Y	Capacity of the customer placing the order.
ExecInst	55	2	ExecInstType	18	Y	Instructions for order handling on exchange.

Field	Offset	Len	Type	Tag Ref Num	Req'd	Description
PegOffsetValue	57	8	Price	211	N	Amount (signed) added to the peg for a pegged order.
PegPriceType	65	1	PegType	1094	N	Defines the type of peg.
ExpireTime	66	8	UTCTimestamp Nanos	126	N	The expiration time of a GoodForTime TimeInForce order. Expire time must be at least a 1 millisecond in the future. Specified in UTC timestamp since unix epoch in nanoseconds.
MinQty	74	4	UINT32	110	N	Minimum quantity of an order to be executed.
DisplayQty	78	4	UINT32	1138	N	Quantity of the order to be displayed.
DisplayMethod	82	1	DispMethodType	1084	N	Defines the replenishment size behavior for a reserve order. This should be null for a fully displayed order.
ReserveReplenish Timing	83	1	ReserveReplenish TimingType	21006	N	Defines the replenishment timing behavior for a reserve order.
DisplayMinIncr	84	4	UINT32	1087	N	Defines the minimum increment to be used when calculating a random refresh of DisplayQty.
LocateReqd	88	1	CHAR	114	N	Required on Short Sell and Short Sell Exempt orders only. Only acceptable value is 'N' which indicates the member has secured the required locate.
RepriceFrequency	89	1	RepriceFrequency Type	21020	N	Defines the frequency of a reprice. If this tag is not sent then the order will not be repriced.
RepriceBehavior	90	1	RepriceBehavior Type	21021	N	Defines the reprice behavior when market is locked or crossed.
CancelGroupID	91	2	UINT16	21000	N	Unique identifier of custom cancel group.
StpGroupID	93	2	StpGroupIDType	2362	N	Unique identifier of custom self trade prevention group.
SelfTradePrevention	95	1	SelfTrade PreventionType	21001	N	Defines the desired behavior in the event of a wash. The UINT8 Null (0xFF) value disables self trade prevention.
RiskGroupID	96	2	UINT16	21005	N	Unique identifier of a custom risk control set to be applied to this order. The UINT16 Null value (0xFFFF) disables custom risk controls.

5.1.2 OrderCancelReplaceRequest

The order cancel/replace request is used to change the parameters of an existing order.

This message corresponds to FIX message type G

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 63 (0x3F), template = 2
OrigClOrdID	6	16	CHAR	41	Y	ClOrdID (11) of the previous order (NOT the initial order of the day) .
ClOrdID	22	16	CHAR	11	Y	Unique identifier of the replacement order as assigned by the client. Uniqueness must be guaranteed within a single trading day.
Symbol	38	6	CHAR	55	Y	This field may not change from the value stated in the original order.
SymbolSfx	44	6	CHAR	65	N	This field may not change from the value stated in the original order.
Side	50	1	SideType	54	Y	Side may only change between Sell and SellShort.
OrderQty	51	4	UINT32	38	Y	Quantity ordered. This represents the number of shares for equities.
OrdType	55	1	OrdType	40	Y	OrderType may only change from Limit order to Market order.
Price	56	8	Price	44	N	Price per unit of quantity (e.g. per share). Changing the price will cause the order to lose its priority in the book.
DisplayQty	64	4	UINT32	1138	N	Quantity of the order to be displayed.
LocateReqd	68	1	CHAR	114	N	Required on Short Sell and Short Sell Exempt orders only. Only acceptable value is 'N' which indicates the member has secured the required locate.

5.1.3 OrderCancelRequest

The order cancel request message requests the cancellation all remaining size on an order.

Note: this message can be used to cancel an order for a different account by specifying the order's exchange order ID (`OrderID`), the remote order may have been entered via MEMO or MEMO FIX supported ports.

This message corresponds to FIX message type F

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 52 (0x34), template = 3
OrigClOrdID	6	16	CHAR	41	Y*	ClOrdID (11) of the previous order (NOT the initial order of the day) as assigned by the customer.
OrderID	22	8	UINT64	37	Y*	OrderID as assigned by the exchange.
ClOrdID	30	16	CHAR	11	Y	An identifier for the cancel request.
Symbol	46	6	CHAR	55	Y	This must be the same symbol as the order request.
SymbolSfx	52	6	CHAR	65	N	Additional information about the security (e.g. preferred, warrants, etc.).

Note: either OrigClOrdID or OrderID fields must be present.

5.1.4 MassCancelRequest

The mass cancel request message requests the cancellation all remaining size on a set of existing orders that match the given criteria.

Note: this message can be used to cancel orders from different accounts. Orders affected by this request may have been entered via either MEMO or MEMO FIX supported protocols.

Field	Offset	Len	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 47 (0x2F), template = 4
ClOrdID	6	16	CHAR	11	Y	A unique identifier for the mass cancel request.
Symbol	22	6	CHAR	55	N	For Mass Cancel scenario, providing a value here will result in all orders for that symbol being canceled.
SymbolSfx	28	6	CHAR	65	N	Additional information about the security (e.g. preferred, warrants, etc.).
Side	34	1	SideType	54	N	Side of order. If Null (`0xFF`) then side is not used as a cancel filter.
Lower ThanPrice	35	8	Price	N/A	N	For Mass Cancel scenario, Any order with a lower or equal limit price would be included in the cancel filter. This filter can only be used in conjunction with a specific symbol.
Higher ThanPrice	43	8	Price	N/A	N	For Mass Cancel scenario, Any order with a higher or equal limit price would be included in the cancel filter. This filter can only be used in conjunction with a specific symbol.
CancelGroupID	51	2	UINT16	21000	N	Identifier of custom cancel group.

5.2 Server To Client

5.2.1 ExecutionReport

The MEMO schema defines multiple Execution Report messages each conforming to FIX message type 8.

5.2.1.1 ExecutionReport_PendingNew

The ExecutionReport_PendingNew is a response to NewOrderSingle indicating the request has been received by the exchange and is in the process of being handled. This message is sent by default and may be disabled upon request.

Field	Offset	Len	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 125 (0x7D), template = 5
SendingTime	6	8	UTCTimestamp Nanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
OrderID	14	8	UINT64	37	Y	OrderID as assigned by the exchange.
ClOrdID	22	16	CHAR	11	Y	As stated in the order.
ExecID	38	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
MPID	46	4	CHAR	21007	Y	The MPID associated with the new order, if it is not supplied, the MPID assigned as the default MPID for the Port/Account is associated with the new order.
OrdStatus	50	1	OrdStatusType	39	Y	The status of the order.
Symbol	51	6	CHAR	55	Y	As stated in the order.
SymbolSfx	57	6	CHAR	65	N	As stated in the order.
Side	63	1	SideType	54	Y	As stated in the order.
OrdType	64	1	OrdType	40	Y	As stated in the order.
OrderQty	65	4	UINT32	38	Y	As stated in the order.
Price	69	8	Price	44	N	As stated in the order.
TimeInForce	77	1	TimeInForceType	59	Y	As stated in the order.
OrderCapacity	78	1	OrderCapacityType	528	Y	As stated in the order.
CustOrderCapacity	79	1	CustOrderCapacity Type	582	Y	As stated in the order.
ExecInst	80	2	ExecInstType	18	Y	As stated in the order.
PegOffsetValue	82	8	Price	211	N	As stated in the order.

Field	Offset	Len	Type	Tag Ref Num	Req'd	Description
PegPriceType	90	1	PegType	1094	N	As stated in the order.
ExpireTime	91	8	UTCTimestampNanos	126	N	As stated in the order.
MinQty	99	4	UINT32	110	N	As stated in the order.
DisplayQty	103	4	UINT32	1138	N	As stated in the order.
DisplayMethod	107	1	DispMethodType	1084	N	As stated in the order.
Reserve ReplenishTiming	108	1	ReserveReplenish TimingType	21006	N	As stated in the order.
DisplayMinIncr	109	4	UINT32	1087	N	As stated in the order.
LocateReqd	113	1	CHAR	114	N	As stated in the order.
RepriceFrequency	114	1	RepriceFrequencyType	21020	N	As stated in the order.
RepriceBehavior	115	1	RepriceBehaviorType	21021	N	As stated in the order.
CancelGroupID	116	2	UINT16	21000	N	As stated in the order.
StpGroupID	118	2	StpGroupIDType	2362	N	As stated in the order.
SelfTrade Prevention	120	1	SelfTrade PreventionType	21001	N	As stated in the order.
RiskGroupID	121	2	UINT16	21005	N	As stated in the order.
LeavesQty	123	4	UINT32	151	Y	Quantity open for further execution.
CumQty	127	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.

5.2.1.2 ExecutionReport_New

The ExecutionReport_New is a response to NewOrderSingle request in the event the order was accepted by the exchange. This message echos back to the requester all the fields as been configured on the request with additional information about the state of the order.

Field	Offset	Len	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 133 (0x85), template = 6
SendingTime	6	8	UTCTimestampNanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
OrderID	14	8	UINT64	37	Y	OrderID as assigned by the exchange.
ClOrdID	22	16	CHAR	11	Y	As stated in the order.
ExecID	38	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
MPID	46	4	CHAR	21007	Y	The MPID associated with the new order, if it is not supplied, the MPID assigned as the default MPID for the Port/Account is associated with the new order.
OrdStatus	50	1	OrdStatusType	39	Y	The status of the order.
Symbol	51	6	CHAR	55	Y	As stated in the order.
SymbolSfx	57	6	CHAR	65	N	As stated in the order.
Side	63	1	SideType	54	Y	As stated in the order.
OrdType	64	1	OrdType	40	Y	As stated in the order.
OrderQty	65	4	UINT32	38	Y	As stated in the order.
Price	69	8	Price	44	N	As stated in the order.
TimeInForce	77	1	TimeInForceType	59	Y	As stated in the order.
OrderCapacity	78	1	OrderCapacityType	528	Y	As stated in the order.
CustOrderCapacity	79	1	CustOrderCapacityType	582	Y	As stated in the order.
ExecInst	80	2	ExecInstType	18	Y	As stated in the order.
PegOffsetValue	82	8	Price	211	N	As stated in the order.
PegPriceType	90	1	PegType	1094	N	As stated in the order.
ExpireTime	91	8	UTCTimestampNanos	126	N	As stated in the order.
MinQty	99	4	UINT32	110	N	As stated in the order.
DisplayQty	103	4	UINT32	1138	N	As stated in the order.

Field	Offset	Len	Type	Tag Ref Num	Req'd	Description
DisplayMethod	107	1	DispMethodType	1084	N	As stated in the order.
ReserveReplenishTiming	108	1	ReserveReplenishTimingType	21006	N	As stated in the order.
DisplayMinIncr	109	4	UINT32	1087	N	As stated in the order.
LocateReqd	113	1	CHAR	114	N	As stated in the order.
RepriceFrequency	114	1	RepriceFrequencyType	21020	N	As stated in the order.
RepriceBehavior	115	1	RepriceBehaviorType	21021	N	As stated in the order.
CancelGroupID	116	2	UINT16	21000	N	As stated in the order.
StpGroupID	118	2	StpGroupIDType	2362	N	As stated in the order.
SelfTradePrevention	120	1	SelfTradePreventionType	21001	N	As stated in the order.
RiskGroupID	121	2	UINT16	21005	N	As stated in the order.
LeavesQty	123	4	UINT32	151	Y	Quantity open for further execution.
CumQty	127	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.
TransactTime	131	8	UTCTimestampNanos	60	Y	The time at which the transaction occurred. UTC timestamp since epoch with nanosecond precision.

5.2.1.3 ExecutionReport_Rejected

The ExecutionReport_Rejected is a response to NewOrderSingle request in the event the order was rejected by the MEMX exchange. This message echos back to the requester all the fields as been configured on the request with additional information about the reject reason.

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 54 (0x36), template = 7
SendingTime	6	8	UTCTimestampNanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
ClOrdID	14	16	CHAR	11	Y	As stated in the order.
ExecID	30	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
OrdStatus	38	1	OrdStatusType	39	Y	The status of the order.
Symbol	39	6	CHAR	55	Y	As stated in the order.
SymbolSfx	45	6	CHAR	65	N	As stated in the order.
LeavesQty	51	4	UINT32	151	Y	Quantity open for further execution.
CumQty	55	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.
RejectReason	59	1	OrderRejectReasonCode	103	Y	Reason code for order rejection.

5.2.1.4 ExecutionReport_Trade

The ExecutionReport_Trade is an unsolicited response triggered as a result of a trade. The message contains all the data describing the trade.

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 79 (0x4F), template = 8
SendingTime	6	8	UTCTimestampNanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
OrderID	14	8	UINT64	37	Y	OrderID as assigned by the exchange.
ClOrdID	22	16	CHAR	11	Y	The unique identifier as specified by the client.
ExecID	38	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
OrdStatus	46	1	OrdStatusType	39	Y	The status of the order.
LastQty	47	4	UINT32	32	Y	Quantity (e.g. shares) bought/sold on this (last) fill.
LastPx	51	8	Price	31	Y	Price of this (last) fill.
LeavesQty	59	4	UINT32	151	Y	Quantity open for further execution.
CumQty	63	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.
TransactTime	67	8	UTCTimestampNanos	60	Y	The time at which the transaction occurred. UTC timestamp since epoch with nanosecond precision.
LastLiquidityInd	75	1	LastLiquidityIndType	851	Y	Indicator denoting whether the referenced order removed liquidity from or added liquidity to the MEMX book.
LastMkt	76	1	ExchangeCode	30	Y	Market of execution for last fill, or an indication of the market where an order was routed.
TrdMatchID	77	8	UINT64	880	Y	Identifier assigned to the Trade by the matching system. In case of executions on orders routed to another exchange, the TrdMatchID field will be filled on a best-effort basis.

5.2.1.5 ExecutionReport_PendingCancel

The ExecutionReport_PendingCancel is a response to OrderCancelRequest request indicating the request has been received by the exchange and is in the process of being handled.

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 77 (0x4d), template = 9
SendingTime	6	8	UTCTimestampNanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
OrderID	14	8	UINT64	37	Y	OrderID as assigned by the exchange.
ClOrdID	22	16	CHAR	11	Y	As stated in the cancel request
OrigClOrdID	38	16	CHAR	41	Y	As stated in the cancel request
ExecID	54	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
Symbol	62	6	CHAR	55	Y	As stated on the previous order
SymbolSfx	68	6	CHAR	65	N	As stated on the previous order
OrdStatus	74	1	OrdStatusType	39	Y	The status of the order.
LeavesQty	75	4	UINT32	151	Y	Quantity open for further execution.
CumQty	79	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.

5.2.1.6 ExecutionReport_Canceled

The ExecutionReport_Canceled message is an unsolicited event on an order triggered by a successful handling of cancel / mass cancel request for that order. Orders that are canceled by the exchange because of Time in Force instructions shall have an OrdStatus as Expired. All other cases shall have OrdStatus as Canceled.

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 74 (0x4A), template = 11
SendingTime	6	8	UTCTimestamp Nanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
ClOrdID	14	16	CHAR	11	Y	The identifier for the cancel request or the canceled order in the event of an unsolicited cancel*
OrigClOrdID	30	16	CHAR	41	N	A Unique identifier for the canceled order if available*
OrderID	46	8	UINT64	37	Y	OrderID as assigned by the exchange to the canceled order.
ExecID	54	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
OrdStatus	62	1	OrdStatusType	39	Y	The status of the order.
LeavesQty	63	4	UINT32	151	Y	Quantity open for further execution.
CumQty	67	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.
CancelReason	71	1	CancelReasonCode	21004	Y	Reason code for order cancellation.
TransactTime	72	8	UTCTimestamp Nanos	60	Y	The time at which the transaction occurred. UTC timestamp since epoch with nanosecond precision.

* Orders that are canceled as a result of a mass cancel request will be reported back to customers as Unsolicited cancel i.e. the `ClOrdID` and `OrigClOrdID` fields will contain the canceled order identifier.

5.2.1.7 ExecutionReport_PendingReplace

The ExecutionReport_PendingReplace message is a response to OrderCancelReplaceRequest request indicating the request has been received by the exchange and is in the process of being handled.

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 96 (0x60), template = 13
SendingTime	6	8	UTCTimestamp Nanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
OrderID	14	8	UINT64	37	Y	OrderID as assigned by the exchange to the canceled order.
ClOrdID	22	16	CHAR	11	Y	As stated in the replace request
OrigClOrdID	38	16	CHAR	41	Y	As stated in the replace request
ExecID	54	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
Symbol	62	6	CHAR	55	Y	As stated on the previous order
SymbolSfx	68	6	CHAR	65	N	As stated on the previous order
Side	74	1	SideType	54	Y	As stated on the previous order
OrderQty	75	4	UINT32	38	Y	As stated on the previous order
OrdType	79	1	OrdType	40	Y	As stated on the previous order
Price	80	8	Price	44	N	As stated on the previous order
DisplayQty	88	4	UINT32	1138	N	As stated on the previous order
LocateReqd	92	1	CHAR	114	N	As stated on the previous order
OrdStatus	93	1	OrdStatusType	39	Y	The status of the order. (PendingReplace)
LeavesQty	94	4	UINT32	151	Y	Quantity open for further execution.
CumQty	98	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.

5.2.1.8 ExecutionReport_Replaced

The ExecutionReport_Replaced is an unsolicited event on an order triggered by a successful handling of OrderCancelReplaceRequest for the original order. The original order can be considered fully cancelled and replaced by the stated replacement order upon receipt of this event.

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 104 (0x68), template = 14
SendingTime	6	8	UTCTimestampNanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
OrderID	14	8	UINT64	37	Y	OrderID as assigned by the exchange.
ClOrdID	22	16	CHAR	11	Y	As stated in the replace request
OrigClOrdID	38	16	CHAR	41	Y	As stated in the replace request
ExecID	54	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
Symbol	62	6	CHAR	55	Y	As stated in the replace request
SymbolSfx	68	6	CHAR	65	Y	As stated in the replace request
Side	74	1	SideType	54	Y	As stated in the replace request
OrderQty	75	4	UINT32	38	Y	As stated in the replace request
OrdType	79	1	OrdType	40	Y	As stated in the replace request
Price	80	8	Price	44	Y	As stated in the replace request
DisplayQty	88	4	UINT32	1138	Y	As stated in the replace request
LocateReqd	92	1	CHAR	114	Y	As stated in the replace request
OrdStatus	93	1	OrdStatusType	39	Y	The status of the order.
LeavesQty	94	4	UINT32	151	Y	Quantity open for further execution.
CumQty	98	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.
TransactTime	102	8	UTCTimestampNanos	60	Y	The time at which the transaction occurred. UTC timestamp since epoch with nanosecond precision.

5.2.1.9 ExecutionReport_TradeCorrection

The ExecutionReport_TradeCorrection is an unsolicited event notifying of a change to a price and/or quantity of a previously reported trade.

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 77 (0x4D), template = 15
SendingTime	6	8	UTCTimestampNanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
OrderID	14	8	UINT64	37	Y	OrderID as assigned by the exchange.
ClOrdID	22	16	CHAR	11	Y	The ClOrdID associated with the order.
ExecID	38	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
ExecRefID	46	8	UINT64	19	Y	The ExecID of the trade being corrected
TrdMatchID	54	8	UINT64	880	Y	Identifier assigned to the Trade by the matching system. In case of executions on orders routed to another exchange, the TrdMatchID field will be filled on a best-effort basis.
OrdStatus	62	1	OrdStatusType	39	Y	Order status
LastPx	63	8	Price	31	Y	Corrected trade price.
LastQty	71	4	UINT32	32	Y	Corrected trade quantity.
LeavesQty	75	4	UINT32	151	Y	Quantity open for further execution.
CumQty	79	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.

5.2.1.10 ExecutionReport_TradeBreak

The ExecutionReport_TradeBreak is an unsolicited event notifying of a cancelation of a previously reported trade.

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 65 (0x41), template = 16
SendingTime	6	8	UTCTimestamp Nanos	52	Y	UTC timestamp since unix epoch with nanosecond precision.
OrderID	14	8	UINT64	37	Y	OrderID as assigned by the exchange.
ClOrdID	22	16	CHAR	11	Y	The ClOrdID associated with the order.
ExecID	38	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
ExecRefID	46	8	UINT64	19	Y	The ExecID of the trade being corrected
TrdMatchID	54	8	UINT64	880	Y	Identifier assigned to the Trade by the matching system. In case of executions on orders routed to another exchange, the TrdMatchID field will be filled on a best-effort basis.
OrdStatus	62	1	OrdStatusType	39	Y	Order status
LeavesQty	63	4	UINT32	151	Y	Quantity open for further execution.
CumQty	67	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.

5.2.1.11 ExecutionReport_Restatement

The ExecutionReport_Restatement is an unsolicited event to notify the client that an open order has been updated by the MEMX system. This message may be sent for orders configured for automatic repricing via the RepriceFrequency instruction. This message can also be sent if the order quantity is partially or fully cancelled because of Self Trade Prevention. This message will not be sent for Pegged Orders.

NOTE: This message will be sent in the case of an order entered with the SingleReprice instruction, when the system is able to display the order at the booking price, if the order was repriced and displayed away from its booking price upon entry. In this case, the lastPx field will be the same as the previous Restatement, to indicate that the position on order book has not changed, but the message still will be sent to allow for correlation with the displayed orders on the MEMOIR Depth feed.

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 70 (0x46), template = 17
SendingTime	6	8	UTCTimestamp Nanos	52	Y	UTC timestamp since unix epoch with nanosecond precision.
OrderID	14	8	UINT64	37	Y	OrderID as assigned by the exchange.
ClOrdID	22	16	CHAR	11	Y	The ClOrdID associated with the order.
ExecID	38	8	UINT64	17	Y	Unique identifier of execution message as assigned by the exchange. Uniqueness is guaranteed within a single trading day.
OrdStatus	46	1	OrdStatusType	39	Y	Order status
LastPx	47	8	Price	31	Y	ExecRestatementType(1): The updated booked price. This is the price that the order can match at. It may be different than the displayed price. ExecRestatementType(2): The match price if this order had not been prevented from executing due to STP restrictions.
LeavesQty	55	4	UINT32	151	Y	Quantity open for further execution.
CumQty	59	4	UINT32	14	Y	Total quantity (e.g. number of shares) filled.
LastShares	63	4	UINT32	32	N	ExecRestatementType(2): The number of shares that would have matched if this order had not been prevented from executing due to STP restrictions.
ExecRestatement Reason	67	1	ExecRestatement Type	378	Y	The reason for the restatement.
TransactTime	68	8	UTCTimestamp Nanos	60	Y	The time at which the transaction occurred. UTC timestamp since epoch with nanosecond precision.

5.2.2 PendingMassCancel

The PendingMassCancel is a response to MassCancelRequest request indicating the request has been received by the exchange and is in the process of being handled.

Field	Offset	Len	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 55 (0x37), template = 10
SendingTime	6	8	UTCTimestamp Nanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
ClOrdID	14	16	CHAR	11	Y	The identifier for this mass cancel, as stated in the mass cancel request.
Symbol	30	6	CHAR	55	N	As stated in the mass cancel request
SymbolSfx	36	6	CHAR	65	N	As stated in the mass cancel request
Side	42	1	SideType	54	N	As stated in the mass cancel request
LowerThanPrice	43	8	Price	N/A	N	As stated in the mass cancel request
HigherThanPrice	51	8	Price	N/A	N	As stated in the mass cancel request
CancelGroupID	59	2	UINT16	21000	N	As stated in the mass cancel request

5.2.3 MassCancelReject

The MassCancelReject message is used by the exchange to indicate a mass cancel request cannot be honored as sent.

Field	Offset	Len	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 56 (0x38), template = 20
SendingTime	6	8	UTCTimestamp Nanos	52	Y	The time the message was sent. UTC timestamp since unix epoch with nanosecond precision.
ClOrdID	14	16	CHAR	11	Y	The identifier for this mass cancel, as stated in the mass cancel request.
Symbol	30	6	CHAR	55	N	As stated in the mass cancel request
SymboSfx	36	6	CHAR	65	N	As stated in the mass cancel request
Side	42	1	SideType	54	N	As stated in the mass cancel request
LowerThanPrice	43	8	Price	N/A	N	As stated in the mass cancel request
HigherThanPrice	51	8	Price	N/A	N	As stated in the mass cancel request
CancelGroupID	59	2	UINT16	21000	N	As stated in the mass cancel request
RejectReason	61	1	MassCancelReject ReasonCode	N/A	Y	Code to identify reason for mass cancel rejection

5.2.4 MassCancelDone

The MassCancelDone is sent when all the orders specified in the MassCancelRequest request have been processed.

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 24 (0x18), template = 12
SendingTime	6	8	UTCTimestamp Nanos	52	Y	The time the ExecutionReport was sent. UTC timestamp since unix epoch with nanosecond precision.
ClOrdID	14	16	CHAR	11	Y	The identifier for this cancel, as stated in the cancel request.

5.2.5 OrderCancelReject

The OrderCancelReject message is used by the exchange to indicate a cancel request or cancel-replace request cannot be honored.

This message corresponds to FIX message type 9

Field	Offset	Length	Type	Tag Ref Num	Req'd	Description
SBE Header	0	6	N/A	N/A	Y	SBE Header with blockLength = 26 (0x1A), template = 18
SendingTime	6	8	UTCTimestamp Nanos	52	Y	UTC timestamp since unix epoch with nanosecond precision.
ClOrdID	14	16	CHAR	11	Y	Unique identifier from the cancel request.
CxlRejResponseTo	30	1	CxlRejResponse ToType	434	Y	Identifies the request that this cancel reject is responding to.
CxlRejReason	31	1	CancelReject ReasonCode	102	Y	Code to identify reason for cancel rejection.

6 Appendices

6.1 Reserve Orders

6.1.1 Display Method Initial

The DisplayQty field specifies the number of shares of the reserve order to display. If the displayed quantity drops below a round lot, then the displayed portion is replenished to display DisplayQty shares.
The DisplayMinInc field should be Null (`0xFFFF`) on the NewOrderSingle.

6.1.1.1 Example

NewOrderSingle with the following fields OrderQty = 2000, DisplayQty = 1000, DisplayMethod = Initial.

The displayed size after replenishment will always be 1000.

Event	Display	Hidden	Filled
Initial Order	1000	1000	0
Trade 950	50	1000	950
Replenish 950	1000	50	950
Trade 1000	0	50	1950
Replenish	50	0	1950

6.1.2 Display Method Random

When Random is selected on a NewOrderSingle, the initial displayed size is as specified by the DisplayQty field. When the size drops below a round lot then the displayed portion of the order is replenished. The new displayed size is determined by randomly choosing a number between (DisplayQty – DisplayMinIncr) and (DisplayQty + DisplayMinIncr). This number is rounded to the nearest round lot and then added to remaining displayed portion of the order. If the replenish size is greater than the remaining size on the order then the remaining size is used as the replenishment amount.

6.1.2.1 Example

NewOrderSingle with the following fields OrderQty = 3000, DisplayQty = 1000, DisplayMinIncr = 600, DisplayMethod = Random.

The replenish size will be randomly selected between (DisplayQty – DisplayMinInc) = 400 and (DisplayQty + DisplayMinInc) = 1600, in 100 share increments (the round lot size).

Event	Display	Hidden	Filled
Initial Order	1000	2000	0
Trade 950	50	2000	950
Random Replenish size 700	750	1300	950
Trade 750	0	1300	1700
Random Replenish size 1500 (only 1300 remains)	1300	0	1700

6.1.3 ReserveReplenishTimingType Random

When Random is selected, the replenish function happens at a random time up to 1 millisecond after the execution that triggered the replenishment. The displayed size after replenishment is the minimum of remaining hidden size and the DisplayQty specified in the NewOrderSingle.

6.1.4 Display Method Undisclosed

Select Undisclosed to make the order completely hidden on the book for the lifetime of the order.

6.2 Mass Cancel Flow

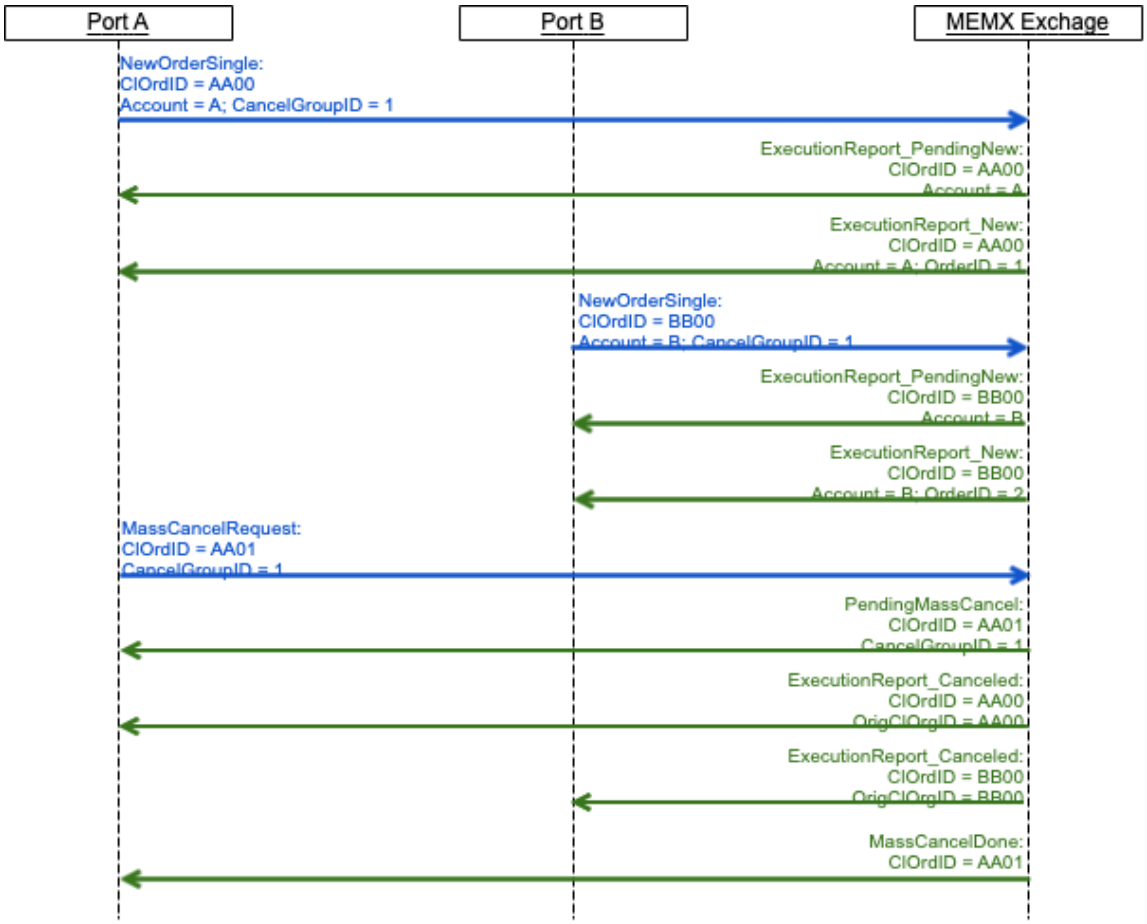
The Mass Cancel feature provides customers with a way to cancel a group of orders belonging to the firm that match a set of parameters provided in the MassCancelRequest message.

The orders matching the set of parameters may come through any port belonging to the customer regardless of which protocol was used to enter them.

Once a MassCancelRequest is sent, the following sequence of messages will follow:

1. A PendingMassCancel response is sent down the requesting port to indicate that the request has been accepted and is being processed by the exchange.
2. ExecutionReport_Canceled messages are sent for all orders matching the requested criteria down the same port they were originally sent through.
3. A MassCancelDone message is sent down the same port which originated the MassCancelRequest to indicate the process has been completed.

The following message diagram depicts this flow:



6.3 Encoding Examples

The below examples illustrate complete encoded messages as they will appear on the wire. Note: the code fragments shown are representative of how to set field values in the SBE messages. Your interface may be be different.

6.3.1 Client to Server Requests:

6.3.1.1 New Order - Limit Order

```
newOrderSingleEncoder.clOrdID("CID0000000001");
newOrderSingleEncoder.mPID("ABCD");
newOrderSingleEncoder.symbol("AAPL");
newOrderSingleEncoder.side(SideType.SellShort);
newOrderSingleEncoder.orderQty(100);
newOrderSingleEncoder.ordType(OrdType.Limit);
newOrderSingleEncoder.price().mantissa(386980000);
newOrderSingleEncoder.timeInForce(TimeInForceType.Day);
newOrderSingleEncoder.orderCapacity(OrderCapacityType.Agency);
newOrderSingleEncoder.custOrderCapacity(CustOrderCapacityType.MemberTradingOnTheirOwnAccount);
newOrderSingleEncoder.execInst().externalRoutingNotAllowed(true);
newOrderSingleEncoder.pegOffsetValue().mantissa(PriceTypeEncoder.mantissaNullValue());
newOrderSingleEncoder.pegPriceType(PegType.NULL_VAL);
newOrderSingleEncoder.expireTime().time(UTCTimestampNanosEncoder.timeNullValue());
newOrderSingleEncoder.minQty(NewOrderSingleEncoder.minQtyNullValue());
newOrderSingleEncoder.displayQty(NewOrderSingleEncoder.displayQtyNullValue());
newOrderSingleEncoder.displayMethod(DispMethodType.NULL_VAL);
newOrderSingleEncoder.reserveReplenishTiming(ReserveReplenishTimingType.NULL_VAL);
newOrderSingleEncoder.displayMinIncr(NewOrderSingleEncoder.displayMinIncrNullValue());
newOrderSingleEncoder.locateReqd(NewOrderSingleEncoder.locateReqdNullValue());
newOrderSingleEncoder.repriceFrequency(RepriceFrequencyType.None);
newOrderSingleEncoder.repriceBehavior(RepriceBehaviorType.NULL_VAL);
newOrderSingleEncoder.cancelGroupId(1);
newOrderSingleEncoder.stpGroupId(2);
newOrderSingleEncoder.selfTradePrevention(SelfTradePreventionType.CancelOldest);
newOrderSingleEncoder.riskGroupId(3);

Hex code:
00000000: 005c0101 01074349 44303030 30303030   |.\....CID0000000|
00000010: 30303100 00004142 43444141 504c0000   |001…ABCDAAPL..|
00000020: 00000000 00000300 00006402 00000000   |……..…….d……..|
00000030: 1710d8a0 01010100 04800000 00000000   |…….............|
00000040: 00ffffff ffffffff ffffffff ffffffff   |…….............|
00000050: ffffffff ffffffff 0003ff00 01000202   |…….............|
00000060: 0003                                  |..|
```

6.3.2 Server to Client Responses:

6.3.2.1 Execution Report – Pending New

```
executionReport_pendingNewEncoder.sendingTime().time(getUTCTimeNanos());
executionReport_pendingNewEncoder.orderID(100000000L);
executionReport_pendingNewEncoder.clOrdID("CID0000000001");
executionReport_pendingNewEncoder.execID(200000000L);
executionReport_pendingNewEncoder.mPID("ABCD");
executionReport_pendingNewEncoder.ordStatus(OrdStatusType.PendingNew);
executionReport_pendingNewEncoder.symbol("AAPL");
executionReport_pendingNewEncoder.side(SideType.SellShort);
executionReport_pendingNewEncoder.orderQty(100);
executionReport_pendingNewEncoder.ordType(OrdType.Limit);
executionReport_pendingNewEncoder.price().mantissa(386980000);
executionReport_pendingNewEncoder.timeInForce(TimeInForceType.Day);
executionReport_pendingNewEncoder.orderCapacity(OrderCapacityType.Agency);
executionReport_pendingNewEncoder.custOrderCapacity(CustOrderCapacityType.MemberTradingOnTheirOwnAccount);
executionReport_pendingNewEncoder.execInst().externalRoutingNotAllowed(true);
executionReport_pendingNewEncoder.pegOffsetValue().mantissa(PriceTypeEncoder.mantissaNullValue());
executionReport_pendingNewEncoder.pegPriceType(PegType.NULL_VAL);
executionReport_pendingNewEncoder.expireTime().time(UTCTimestampNanosEncoder.timeNullValue());
executionReport_pendingNewEncoder.minQty(NewOrderSingleEncoder.minQtyNullValue());
executionReport_pendingNewEncoder.displayQty(NewOrderSingleEncoder.displayQtyNullValue());
executionReport_pendingNewEncoder.displayMethod(DispMethodType.NULL_VAL);
executionReport_pendingNewEncoder.reserveReplenishTiming(ReserveReplenishTimingType.NULL_VAL);
executionReport_pendingNewEncoder.displayMinIncr(NewOrderSingleEncoder.displayMinIncrNullValue());
executionReport_pendingNewEncoder.locateReqd(NewOrderSingleEncoder.locateReqdNullValue());
executionReport_pendingNewEncoder.repriceFrequency(RepriceFrequencyType.None);
executionReport_pendingNewEncoder.repriceBehavior(RepriceBehaviorType.NULL_VAL);
executionReport_pendingNewEncoder.cancelGroupId(1);
executionReport_pendingNewEncoder.stpGroupId(2);
executionReport_pendingNewEncoder.selfTradePrevention(SelfTradePreventionType.CancelOldest);
executionReport_pendingNewEncoder.riskGroupId(3);
executionReport_pendingNewEncoder.leavesQty(100);
executionReport_pendingNewEncoder.cumQty(0);

Hex code:
00000000: 007d0501 01070001 c93f5a28 a9c60000   |.}……..?Z(….|
00000010: 000005f5 e1004349 44303030 30303030   |……CID0000000|
00000020: 30303100 00000000 00000beb c2004142   |001……..….AB|
00000030: 43440741 41504c00 00000000 00000003   |CD.AAPL……..|
00000040: 02000000 64000000 001710d8 a0010101   |….d……….|
00000050: 00048000 00000000 0000ffff ffffffff   |…….............|
00000060: ffffffff ffffffff ffffffff ffffffff   |…….............|
00000070: ff0003ff 00010002 02000300 00006400   |…….…….d.|
00000080: 000000                                |…|
```

7 Revision History

Revision	Date	Change
	Feb 10, 2022	Initial Revision
B	Aug 8, 2022	Added ImmediateMidpointRemoveOnEntry and RetailImmediateRemoveOnEntry values in LastLiquidityIndType
C	Nov 17, 2022	Added AddDisplayedPriceImprovement, AddHiddenPriceImprovement, RetailAddDisplayedPriceImprovement, RetailAddHiddenPriceImprovement values in LastLiquidityIndType