Cboe Europe
TRF FIX Specification

Version 1.32
28 February 2019

Contents

---

1 Overview

This document describes the Cboe interpretation and implementation of the FIX specification for entry of trade reports into the Cboe Europe Trade Reporting Facility ("TRF").

It is assumed that the reader is familiar with the FIX protocol as described by the FIX Protocol Organisation. Cboe supports FIX 4.2 or FIX 4.4 sessions, and a subset of the applicable FIX application messages for trade reporting.

TRF FIX sessions specified by this document allow participants to:

- Enter OTC trade reports using the 'Trade Capture Report' message.
- Enter and amend Systematic Internalizer one-sided or two-sided quotes using the 'Quote' message.
- Cancel SI quotes using the 'Quote Cancel' message, either on a per-symbol basis, or for all current quotes with a single message.
- Enter SI trade reports, using the 'Trade Capture Report' message.
- Enter MTF trade reports, using the 'Trade Capture Report' message.
- Receive technical and application level acknowledgements for trade reports via the Trade Capture Report Ack and Trade Capture Report messages.
- Receive application level acknowledgements for SI quotes, quote amendments and quote cancellations via the 'Quote Status Report' message.

1.1 Hours of Operation

Refer to the Cboe website for hours of operation.

1.2 Timestamps

All FIX timestamps are GMT as per the FIX standard. Participants are expected to synchronise their clocks with an external time source.

1.3 Symbology

Cboe accepts three symbologies: MTF Common Symbology (UMTF), RIC and ISIN. Different symbologies may be used on different trade reports, but it is recommended that Participants use the same symbology for all.

If using UMTF to identify a symbol, the Participant:

- **must** set *Symbol* (55) to the UMTF symbol;
- *may optionally* set the *SecurityExchange* (207); and,
- *may optionally* set the *Currency* (15).

If using ISIN to identify a symbol, the Participant:

- **must** set *IDSource* (22) to ISIN (4);
- **must** set *SecurityID* (48) to the ISIN;
- *may optionally* set *SecurityExchange* (207) to note the market in which the ISIN trades;
- **must** set the *Currency* (15) field to identify the currency in which the stock is traded; and,
- *may optionally* set the *Symbol* (55).

If using RIC to identify a symbol, the Participant:

- **must** set *IDSource* (22) to RIC (5);
- **must** set *SecurityID* (48) to the RIC;
- *may optionally* set the *SecurityExchange* (207);

- *may optionally* set the *Currency* (15) field; and,
- *may optionally* set the *Symbol* (55).

If using ISIN or RIC to identify a symbol in *SecurityID* (48), and opting to also send *Symbol* (55), the *Symbol* (55) may be specified as the UMTF symbol, the *SecurityID* (48), the RIC or the Ticker code.

A RIC in either *SecurityID* (48) or *Symbol* (55), may be supplied as either a Cboe or primary market RIC listed in the Cboe symbol refernece files for that day. Note that the symbol suffix of the Cboe RIC is not used to determine which APA the trade is being reported to; this is determined by jurisdiction of the submitting bank code.

When specifying an optional value as noted above, the value specified must match the value in Cboe' symbol database. Otherwise, the order will be rejected (subject to notes on "unknown symbol handling" below).

Acknowledgement messages will always respond with the same symbology as was sent in the corresponding New Order Single message.

For trades reported using UMTF or RIC to identify a symbol, that symbol must be known to Cboe, otherwise the trade report will be rejected. For trades reported using ISIN symbology, should the symbol be unknown to Cboe, the trade report will be accepted and enter "unknown symbol handling".

Should the ISIN be known to Cboe, but the combination of symbol and currency not be known to Cboe, the trade report will be accepted and enter "unknown symbol handling".

Given a major currency for an known ISIN listed in a minor currency the currency will be automatically converted and the price scaled as appropriate.

For symbols that have more than one ISIN/currency combination the preferred listing will be used for OTC and SI trade reports.

Trade reports entering "unknown symbol handling" will not be distributed via market data, but will be made widely available from our website.

For additional information about Cboe symbology, see the Cboe Europe Market Guide.

1.4   Tick Sizes

SI Quotes and Trade Capture Reports do not need their price to be on a tick size boundary. For trade reports, should the price specified exceed seven decimal places, it will be truncated to such.

1.5   Assisted Reporting Configuration

The Assisted Reporting model permits submitting firms to enter trade reports on behalf of their clients to address the scenario when the trade reporting obligation falls upon the underlying client. The Assisted Reporting model requires prior arrangement with Cboe and the use of an agreed code for each underlying client for the model to work successfully. Assisted reporting is achieved in the messaging interface by placing the pre-agreed code into the *PartyID* (448) tag in the Trade Capture Report message. For more information regarding Assisted Reporting, please contact your Cboe account manager.

2 FIX Session Protocol

Cboe TRF sessions can use either FIX 4.2 or FIX 4.4 session protocols. The Participant will be provided with a *SenderCompID* (49) and *SenderSubID* (50) that must be sent on every message. The *TargetCompID* (56) for all messages the Participant sends will be BATS, while the *TargetSubID* (57) is TEST for the Cboe test system and PROD for the Cboe production system. All messages the Participant receives will have the sender and target fields swapped, as per the FIX specification.

The following session messages are supported in both directions:

Message	Type	Comment
Logon	A	Begin session (or resume a broken session)
Heartbeat	0	
Test Request	1	
Resend Request	2	
Reject	3	Malformed message or improper session level handling
Sequence Reset	4	Both Gap Fill (*GapFillFlag* (123) = Y) and Reset
Logout	5	used to gracefully close session

2.1 Sequence Numbers

Sequence numbers, both inbound and outbound, will be reset to one each night during the down time.

Messages are processed in sequence order. Behind sequence messages (other than Sequence Reset — Reset) cause immediate logout. Ahead of sequence messages (other than a Resend Request) trigger a message recovery via a Resend Request.

2.2 Logon

The logon must be the first message sent by the Participant after the TCP connection is established. *Encrypt-Method* (98) is ignored (FIX level encryption is not supported).

The IP address of the Participant, the *SenderCompID* (49), *SenderSubID* (50), *TargetCompID* (56) (BATS) and *TargetSubID* (57) (TEST or PROD) will be validated. If validation fails, the connection will be dropped without a reject (to avoid corrupting the Participant's sequence in the case that the Participant merely mistakenly connected to the wrong port).

If the connection is unexpectedly broken, upon reconnection, the Participant may receive a login reply with a sequence number greater than expected. This means that in-flight messages were missed (likely important execution reports). The Participant should issue a Resend request to retrieve the missed messages.

Similarly, Cboe will issue a Resend Request to the Participant for messages that it missed. The Participant may wish to send gap fill messages in place of new orders to avoid submission of potentially stale orders.

HeartBtInt (108) must be specified by the Participant in the Logon message. This value will be clamped between five and 300 seconds and returned in the Logon reply message. We recommend using as low a value as the reliability and latency of your telecommunications channel will allow.

The Cboe FIX 4.4 session protocol does not implement the optional support for 'Logon Message NextExpectedMsgSeqNum Processing'.

2.3 Heartbeat

A Heartbeat message should be sent if the agreed upon *HeartBtInt* (108) has elapsed since the last message sent. If any message has been sent during the preceding *HeartBtInt* (108), a Heartbeat message need not be sent.

2.4 Test Request

If *HeartBtInt* + 1 seconds have elapsed since the last message received, a Test Request should be issued. If another *HeartBtInt* + 1 seconds go by without receiving a message, the TCP connection should be dropped. This ensures that a broken TCP connection will be detected even if the TCP stack doesn't notice (this has been observed to happen in WAN environments, particularly when a VPN is involved).

2.5 Resend Request

A Resend Request message should be processed even if it is received ahead of sequence. Only after resending the requested range (all marked *PossDup* (43) = Y), including any gap fills) should Resend Request be issued in the opposite direction.

As discussed in the FIX 4.2 specification, it is possible to send an open or closed sequence range in a Resend Request (an open range uses sequence zero as the *EndSeqNo* (16)). Cboe will honor either type of request, but will always issue Resend Requests with a closed sequence range.

2.6 Reject

Session level rejects (*MsgType* (35) = 3) are used to indicate violations of the session protocol, or missing (or mangled) fields. These are to be expected during development and certification while the Participant's systems are being adapted for Cboe, but should be extremely rare in production. Application layer rejects (via Trade Capture Ack, Trade Capture Report or Quote Status Report) are normal and should be handled separately. See FIX Application Messages - Cboe to Participant (p. 18) for more details.

2.7 Sequence Reset

Sequence Reset — Gap Fill messages (*GapFillFlag* (123) = Y) must be received in sequence. Any messages (including Gap Fills) sent in response to a Resend Request should have *PossDup* (43) = Y.

Sequence Reset — Reset (*GapFillFlag* (123) ≠ Y) is used only as a last resort, and always by human intervention, to allow an otherwise hopelessly confused session to be resumed. In these cases, all chances at automatic message recovery are lost.

2.8 Logout

Either side may issue a logout to gracefully close the session. The side that issues the logout should process messages normally until it sees the logout reply, and then break the TCP connection. Cboe will typically only request logout after the scheduled end of FIX session.

3 Standard FIX Message Header and Trailer

3.1 Header

Tag	Name	Description
8	*BeginString*	`FIX.4.2` or `FIX.4.4` Must be the first field in the message.
9	*BodyLength*	Length of message following *BodyLength* field up to and including the delimiter preceding the *CheckSum* (10) field. Must be the second field in the message.
35	*MsgType*	Must be the third field in the message.
34	*MsgSeqNum*	Sequential sequence number for the session.
43	*PossDupFlag*	Indicates a message resent from the admin level (has a duplicate sequence number). Defaults to `N`.
49	*SenderCompID*	ID of sender. Assigned by Cboe for messages sent to Cboe. (*TargetCompID* (56) for messages from Cboe.)
52	*SendingTime*	GMT date and time that message was sent. Microsecond level resolution.
56	*TargetCompID*	ID of destination. `BATS` for messages sent to TRF ports. (*SenderCompID* (49) for messages from Cboe.)
57	*TargetSubID*	Sub ID of destination. `TEST` for messages sent to the Cboe test system. `PROD` for messages sent to the Cboe production system. (*SenderSubID* (50) for messages from Cboe.)
97	*PossResend*	Possible resend flag. Cboe does not support *PossResend* (97) = `Y` on the TRF system.
122	*OrigSendingTime*	For messages with *PossDupFlag* (43) = `Y`, indicates time that message was first sent. Microsecond level resolution.
128	*DeliverToCompID*	Service bureau use. Identifies end-client on message **from** Cboe. Must be Cboe approved identifier.

3.2 Trailer

Tag	Name	Description
10	*CheckSum*	Modulo 256 checksum of all characters in the message up to and including the delimiter preceding the *CheckSum* field. Three digits with leading zeroes if necessary.

4 FIX Application Messages — Participant to Cboe

4.1 Trade Capture Report

The Trade Capture Report is used to submit trade reports in a number of scenarios, including:

- Trade reports for the APA service.
- Systematic Internalizer trade reports submitted under the "SI" service.
- Dark MTF trade reporting.

The model supported is as described in the FIX 5.0 (SP2) specification in the *Two-Party Reporting* workflow diagram of the Trade Capture Reporting section, for the specific case where there is a single counterparty.

Whilst we make use of FIX 4.4 and FIX 5.0 messages/tags, these are handled as extensions and operate over either a FIX 4.2 or FIX 4.4 session.

When a new trade report is accepted, a *TradeID* is returned in the corresponding Trade Capture Report confirmation messages. Where applicable, such trade reports may be cancelled, amended or released by specifying the *TradeID*.

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = AE
15	*Currency*	Required if *IDSource* (22) = 4 (ISIN). If *Currency* (15) is included when other symbology is used, it must match the currency expected by Cboe for the given symbol.
18	*ExecInst*	This field must not be supplied on trade reports to the TRF.
22	*IDSource*	Values supported by the TRF: 4 = ISIN 5 = RIC Required if *Symbol* (55) is not set.
31	*LastPx*	Traded price. A value of zero or an indicative price may be used if the price is pending, as denoted by *TradePriceCondition*. This tag may be excluded if using *GrossTradeAmt*.
32	*LastQty*	Quantity (eg. shares) traded.
48	*SecurityID*	ISIN, or RIC if *IDSource* (22) is set.
55	*Symbol*	Security symbol. See Symbology (p. 4) for additional notes.
60	*TransactTime*	Optional, when *TradeReportTransType* (487) = 0. Indicates the date and time the trade took place. Microsecond level resolution.
75	*TradeDate*	Optional. If specified, must match the date component of *TransactTime* (60).
150	*ExecType*	Must be F = Trade
207	*SecurityExchange*	Used when *IDSource* (22) = 4 (ISIN) to identify a specific market. If present must be a valid Reuters exchange code or ISO MIC code. For APA trade reports, may optionally be left blank to use the primary line as designated by Cboe.
381	*GrossTradeAmt*	Total amount traded, expressed in units of currency. A value of of zero or an indicative total amount traded price may be used if the price is pending, as denoted by TradePriceCondition. This tag is only considered when *LastPx* is not specified.

487	TradeReportTransType	Specifies whether this trade report is new, a cancellation, an amendment or a release. Trades reported under the APA service may be cancelled or amended by the participant. Trades that are currently being delayed from publication can be released for immediate publication. Defaults to 'new' if unspecified.
		0 = New
		1 = Cancel
		2 = Replace
		3 = Release
571	TradeReportID	Day-unique ID chosen by client. Cboe will enforce port level day-uniqueness.
		20 characters or less. Characters in ASCII range 33–126 are allowed, except for comma, semicolon, and pipe.
		If the TradeReportID matches a live trade report (one that has been acked, but not confirmed or declined), it will be rejected as duplicate.
574	MatchType	Where VenueType (1430) = 0 (off book), this field models the MMT Level 2 'Trading Mode' field, and takes the following values:
		1 = Trade Reporting (Off-Exchange)
		3 = Trade Reporting (On-Exchange)
		9 = Trade Reporting (Systematic Internalizer)
828	TrdType	This field corresponds to the MMT Level 3.1 field 'Transaction Category'.
		0 = Regular Trade (when none of the following applies)
		62 = Dark Trade
		If TradePriceCondition (1839) = 14 (Trade with Price Improvement) is specified, then the value specified in TrdType (828) will be ignored. Trade with Price Improvement takes precedence above the values listed here.
829	TrdSubType	This optional field corresponds to the MMT Level 3.3 field 'Crossing Trade Indicator' (ACTX). Agency Cross trades may be indicated by setting TrdSubType (829) = 37. Other values are invalid.
855	SecondaryTrdType	This optional field corresponds to the MMT Level 3.5 field 'Benchmark Indicator' (BENC). Benchmark trades may be indicated by setting SecondaryTrdType (855) = 64. Other values are invalid.
856	TradeReportType	Must be 0 (Submit)
1003	TradeID	Used to specify a previously reported trade to be amended, cancelled or released. Mandatory when TradeReportTransType (487) = 1, 2 or 3, must be absent when TradeReportTransType (487) = 0.
1123	TradeHandlingInstr	Must be 1 (Two-Party Report)

1390	*TradePublishIndicator*	This field corresponds to the MMT Level 4.1 field 'Publication Mode', and can be used by the participant to request that the publication be delayed. It is ignored if the trade does not qualify for delayed publication. This field may not be amended, however trades currently being delayed may be released prior to their maximum delay duration using *TradeReportTransType* (487) = 3. Supported values: 0 = Do Not Publish 1 = Publish trade (immediately) 2 = Deferred Publication
1430	*VenueType*	Indicates the type of venue on which this trade occurred. This field models the MMT Level 1 field 'Market Mechanism', and has the following possible values: B = Central Limit Order Book Q = Quote Driven Market D = Dark Order Book O = Off Book A = Periodic Auctions N = Request for Quotes H = Hybrid Market
1838	*NoTradePriceConditions*	If present, indicates the number of *TradePriceCondition* (1839) fields.
1839	*TradePriceCondition*	Optional. Used to indicate values in MMT v3 levels 3.1, 3.6 and 3.8 For MMT Level 3.1 'Transaction Category': 14 = Trade with Price Improvement (RPRI). *TrdType* (828) = 0 should be set. For MMT Level 3.6 'Special Divided Indicator', supported values are: 0 = Cum Dividend (deprecated) 2 = Ex Dividend (deprecated) 13 = Special Dividend (SDIV) For MMT Level 3.8 'Contribution to Price Formation or the Price Discovery Process', supported values are: 16 = Trade not Contributing to the Price Discovery Process (TNCP) 17 = Price is Pending (PNDG)
2405	*ExecMethod*	Optional. Used to indicate that the method by which the trade was executed. This field corresponds to the proposed MMT Level 3.7 'Offbook Automated Liquidity Indicator'. The following values are supported: 0 = Unspecified (default) 1 = Manual 2 = Automated

2667	*AlgorithmicTradeIndicator*	Indicates that the submitted trade was a result of an investment firm engaging in algorithmic trading. Optional.
		0 = No algorithm was involved (the default). 1 = The trade was an algorithmic trade (ALGO).
8013	*TrdRegPublication Reasons*	Optional, to indicate waiver used for the trade. Valid values are: 3 = Reference Price (Dark Book) (for MTF only) (RFPT) 4 = Pre-Trade Transparency Waiver for Illiquid Instrument (for SI only) (ILQD) 5 = Pre-Trade Transparency Waiver for Above Standard Market Size (for SI only) (SIZE) 6 = Deferral for Large in Scale (LRGS) 7 = Deferral for Illiquid Instrument (for RTS2 only) (ILQD) 8 = Deferral for Size Specific (for RTS2 only) (SIZE) Pre-Trade Transparency Waivers ILQD and SIZE can be requested individually or together by specifying one ore more value separated by a space (e.g. 4 5). Based upon what is requested, the system calculates which of these are valid. The business confirmation contains the waiver(s) that have been appplied. Deferrals reasons are requested by using the above values in *TrdRegPublicationReasons*. Deferrals that can be applied together can be specified by separating them by a space (e.g. 6 7). Based upon what is requested, the system calculcates which of these are valid. The business confirmation contains the deferral(s) that have been applied.
552	*NoSides*	Indicates the number of instances of the repeating group *TrdCapRptSideGrp* to follow.

		Repeating Group *TrdCapRptSideGrp* must occur the number of times specified in *NoSides* (552)	
54	*Side*	Must be first field in repeating-group 1 = Buy 2 = Sell 8 = Cross	
1	*Account*	Optional. Reflected back on Trade Capture Report confirmed and declined messages. 16 characters or less (ASCII 33–126).	
447	*PartyIDSource*	Must always be D (Proprietary / Custom Code)	
452	*PartyRole*	Specifies the role of the party to the trade. At this time, this field is mandatory on trades reported to the TRF, and must be specified as *PartyRole* (452) = 7.	
453	*NoPartyIDs*	Must always be 1	
448	*PartyID*	The Cboe Participant ID (4 uppercase letters).	
528	*OrderCapacity*	Optional. A = Agency P = Principal (default) R = Riskless	
625	*TradingSessionSubID*	Models the MMT Level 2 'Trading Mode' field for scenarios where *VenueType* (1430) is not 'O' (off book). Use *MatchType* (574) instead, where *VenueType* (1430) = 0. The following are valid values: 2 = Scheduled Opening Auction 4 = Scheduled Closing Auction 6 = Scheduled Intraday Auction 8 = Unspecified Auction 9 = Unscheduled Auction 3 = Continuous Trading 5 = Post Trading 10 = Out of Main Session Trading The tag must be present on the first side, but is optional in any further sides. If specified in further sides, the values used must be identical to that used in the first.	

	Standard Message Trailer	

4.2 Example Trade Reporting Messages

4.2.1 OTC Trade Reporting

Below illustrates an example of a standard off-book trade report. Firm ABCD is reporting an OTC sell to an unspecified party.

- *BeginString* (8) = FIX.4.4
- *BodyLength* (9) = 100 [1]
- *MsgType* (35) = AE - Trade Capture Report
- *MsgSeqNum* (34) = 7
- *SenderCompID* (49) = ABCD - assigned by Cboe
- *TargetCompID* (56) = BATS
- *SenderSubID* (50) = 0014 - assigned by Cboe
- *TargetSubID* (57) = TEST
- *TradeReportID* (571) = 1234
- *TradeReportTransType* (487) = 0
- *TradeReportType* (856) = 0
- *VenueType* (1430) = 0 - Off Book
- *MatchType* (574) = 1 - privately negotiated (off-book) trade
- *TrdType* (828) = 0 - Regular Trade
- *TradeHandlingInstr* (1123) = 1
- *Currency* (15) = GBX
- *IDSource* (22) = 4
- *SecurityID* (48) = GB012345678
- *SecurityExchange* (207) = L
- *LastQty* (32) = 5500
- *LastPrice* (31) = 123
- *NoSides* (552) = 1
- *Side* (54) = 2 - Sell
- *NoPartyIDs* (453) = 1
- *PartyID* (448) = ABCD
- *PartyRole* (452) = 7
- *CheckSum* (10) = 234 [1]

4.2.2 Dark MTF trade reports

Trade reports submitted by a regulated MTF operating a dark pool are expected to identify themselves as dark trades using *TrdType* (828) = 62. Again, only one instance of *TrdCapRptSideGrp* is applicable, with *Side* (54) = 8 representing a cross, and *PartyID* (448) identifying the Dark MTF.

- *VenueType* (1430) = D - Dark Order Book
- *TrdType* (828) = 62 - Dark Trade
- *NoSides* (552) = 1
- *Side* (54) = 8 - Cross
- *NoPartyIDs* (453) = 1
- *PartyID* (448) = ABCD - representing the Dark MTF
- *PartyRole* (452) = 7
- *TradingSessionSubID* (625) = 3 - Continuous Trading

---

[1]In these examples, *BodyLength* and *Checksum* example values have not been accurately computed.

4.2.3 Systematic Internaliser Trade Reporting

SI trades are identified with *MatchType* (574) = 9. Such trade reports need only contain a single *TrdCapRpt-SideGrp*, confirming the systematic internaliser as the party, and specifying *Side* (54) = 8 to indicate a cross.

- *VenueType* (1430) = 0 - Off Book
- *MatchType* (574) = 9 - Systematic Internalizer
- *TrdType* (828) = 0 - Regular Trade
- *NoSides* (552) = 1
- *Side* (54) = 8 - Cross
- *NoPartyIDs* (453) = 1
- *PartyID* (448) = ABCD - representing the Systematic Internaliser
- *PartyRole* (452) = 7

4.3 Quote

The Quote message may be used by Systematic Internalisers to maintain their quotes under their MiFID pre-trade transparency obligations.

Subsequent Quote messages may be sent to amend the firms quotes. Each Quote message should include a new, day-unique *QuoteID* field. A one-sided quote may be submitted by sending (for example) *BidPx* (132) = 0 and *BidSize* (134) = 0. A Quote which omits the *BidPx* and *BidSize* fields will leave the bid-side quote unchanged. Sending *BidPx* without sending *BidSize*, or vice versa, is invalid (even if the included field is sent with value 0).

Quotes may be cancelled by sending a QuoteCancel message, or a Quote message with zero sent for both price and size.

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = S
117	*QuoteID*	Mandatory unique ID chosen by client for a Quote or QuoteCancel. 18 characters or less. Characters in ASCII range 33–126 are allowed, except for comma, semicolon, and pipe. **Note: Cboe only enforces the uniqueness of QuoteID values among currently live quotes. However, we strongly recommend that you keep your QuoteID values day unique.**
55	*Symbol*	Security symbol. See Symbology (p. 4) for additional notes.
48	*SecurityID*	ISIN, or RIC if *IDSource* (22) is set.
22	*IDSource*	Values supported by the TRF: 4 = ISIN 5 = RIC Required if *Symbol* (55) is not set.
207	*SecurityExchange*	Used when *IDSource* (22) = 4 (ISIN) to identify a specific market. If present must be a valid Reuters exchange code or ISO MIC code. For APA trade reports, may optionally be left blank to use the primary line as designated by Cboe.
132	*BidPx*	Specifies the bid quote price. If omitted, indicates the bid quote is not to be changed, and *BidSize* (134) must also be omitted. If *BidPx* (132) = 0, indicates the bid quote is to be cancelled, and *BidSize* (134) = 0 must also be present.
133	*OfferPx*	Specifies the offer quote price. If omitted, indicates the offer quote is not to be changed, and *OfferSize* (135) must also be omitted. If *OfferPx* (133) = 0, indicates the offer quote is to be cancelled, and *OfferSize* (135) = 0 must also be present.
134	*BidSize*	The quantity of the bid quote. Must be zero when cancelling a quote.
135	*OfferSize*	The quantity of the offer (ask) quote. Must be zero when cancelling a quote.
15	*Currency*	Required if *IDSource* (22) = 4 (ISIN). If *Currency* (15) is included when other symbology is used, it must match the currency expected by Cboe for the given symbol.
	Standard Message Trailer	

4.4 QuoteCancel

The QuoteCancel message may be used to cancel a previously accepted Quote message. One QuoteCancel message must be sent for each symbol which requires quotes to be cancelled. Alternatively, all quotes can be cancelled in one message, using *QuoteCancelType* (298) = 4. QuoteCancel messages should contain a new *QuoteID*, unique for the day across all Quote and QuoteCancel messages.

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = Z
117	*QuoteID*	Mandatory unique ID chosen by client for a Quote or QuoteCancel. 18 characters or less. Characters in ASCII range 33–126 are allowed, except for comma, semicolon, and pipe. **Note: Cboe only enforces the uniqueness of QuoteID values among currently live quotes. However, we strongly recommend that you keep your QuoteID values day unique.**
298	*QuoteCancelType*	Supported values: 1 = Cancel for Symbol 4 = Cancel for All Quotes This field is optional, and defaults to 1. When *QuoteCancelType* (298) = 4, the request will be acknowledged with one QuoteStatusReport message for each existing quote.
295	*NoQuoteEntries*	Must be 1 if *QuoteCancelType* (298) = 1 (Symbol), 0 if *QuoteCancelType* (298) = 4 (All).
Repeating Group *QuoteCancelRptGrp* must occur the number of times specified in *NoQuoteEntries* (295)		
55	*Symbol*	Security symbol. See Symbology (p. 4) for additional notes.
48	*SecurityID*	ISIN, or RIC if *IDSource* (22) is set.
22	*IDSource*	Values supported by the TRF: 4 = ISIN 5 = RIC Required if *Symbol* (55) is not set.
207	*SecurityExchange*	Used when *IDSource* (22) = 4 (ISIN) to identify a specific market. If present must be a valid Reuters exchange code or ISO MIC code. For APA trade reports, may optionally be left blank to use the primary line as designated by Cboe.
15	*Currency*	Required if *IDSource* (22) = 4 (ISIN). If *Currency* (15) is included when other symbology is used, it must match the currency expected by Cboe for the given symbol.
	Standard Message Trailer	

5 FIX Application Messages — Cboe to Participant

5.1 Trade Capture Report Ack

The Trade Capture Report Ack is sent by Cboe to acknowledge the receipt of a Trade Capture Report. It is a technical-level ack, the Trade/Cancel/Amend/Release is not considered to have fully succeeded until a Trade Capture Report is sent with *TradeReportType* (856) of 2 (Accept).

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = AR
15	*Currency*	Copied from the incoming `TradeCaptureReport`, if present.
22	*IDSource*	Copied from the incoming `TradeCaptureReport`, if present.
31	*LastPx*	Copied from the incoming `TradeCaptureReport`, if present. If *GrossTradeAmt* was used instead of *LastPx*, the value here will be indicative. Price is adjusted for allowed precision.
32	*LastQty*	Copied from the incoming `TradeCaptureReport`.
48	*SecurityID*	Copied from the incoming `TradeCaptureReport`, if present.
55	*Symbol*	Copied from the incoming `TradeCaptureReport`, if present.
58	*Text*	If present, indicates reason for the message. Format is one letter reason code followed by colon and space followed by free form text message. Reason codes are: A = Admin D = Duplicate *TradeReportID* H = Halted Y = Symbol Not Supported Z = Unforseen Reason m = Market Access Risk Limit Exceeded o = Max Open Trade Count Exceeded
60	*TransactTime*	Copied from the incoming `TradeCaptureReport`.
75	*TradeDate*	Copied from the incoming `TradeCaptureReport`.
150	*ExecType*	Copied from the incoming `TradeCaptureReport`.
207	*SecurityExchange*	Copied from the incoming `TradeCaptureReport`, if present.
381	*GrossTradeAmt*	Copied from the incoming `TradeCaptureReport`, if present.
487	*TradeReportTransType*	Copied from the incoming `TradeCaptureReport`.
571	*TradeReportID*	Copied from the incoming `TradeCaptureReport`.
572	*TradeReportRefID*	Unique identifier for the trade report as provided by Cboe
574	*MatchType*	Copied from the incoming `TradeCaptureReport`.
828	*TrdType*	Omitted if *TradePriceCondition* (1839) = 14 (Trade with Price Improvement) is set. Otherwise, copied from the incoming `TradeCaptureReport`.
829	*TrdSubType*	Copied from the incoming `TradeCaptureReport`.
855	*SecondaryTrdType*	Copied from the incoming `TradeCaptureReport`.
856	*TradeReportType*	Copied from the incoming `TradeCaptureReport`.
939	*TrdRptStatus*	Will be 0 (Accepted) or 1 (Rejected)
1003	*TradeID*	Copied from the incoming `TradeCaptureReport`, if present.
1123	*TradeHandlingInstr*	Copied from the incoming `TradeCaptureReport`.
1390	*TradePublishIndicator*	Copied from the incoming `TradeCaptureReport`.
1430	*VenueType*	Copied from the incoming `TradeCaptureReport`.
1838	*NoTradePriceConditions*	Copied from the incoming `TradeCaptureReport`.

Repeating Group *TradePriceConditionGrp* must occur the number of times specified in *NoTradePriceConditions* (1838)			
	...	...	Entire block copied from incoming TradeCaptureReport, although order may be adjusted
2405	*ExecMethod*		Copied from the incoming `TradeCaptureReport`, if present.
2667	*AlgorithmicTradeIndicator*		Copied from the incoming `TradeCaptureReport`, if present.
8013	*TrdRegPublicationReasons*		Copied from the incoming `TradeCaptureReport`, if present.
552	*NoSides*		Copied from the incoming `TradeCaptureReport`.
Repeating Group *TrdCapRptSideGrp* must occur the number of times specified in *NoSides* (552)			
	...	...	Entire block copied from incoming TradeCaptureReport
9688	*OrigCompID*		**Drop only.** *TargetCompID* (56) of original FIX TradeCaptureReport from Cboe to the Participant. Drop port must be configured to send this optional field.
9689	*OrigSubID*		**Drop only.** *TargetSubID* (57) of original FIX TradeCaptureReport from Cboe to the Participant. Drop port must be configured to send this optional field.
	Standard Message Trailer		

5.2 Trade Capture Report

The 'Trade Capture Report' is sent from Cboe to the participant in order to confirm that a Trade Capture Report has been fully processed. It is a business-level confirmation as distinct from the technology level acknowledgement sent as a 'Trade Capture Report Ack'.

Tag	Name	Description
	Standard Message Header	MsgType (35) = AE
15	Currency	Copied from the incoming TradeCaptureReport, if present.
22	IDSource	Copied from the incoming TradeCaptureReport, if present.
31	LastPx	Traded Price.
32	LastQty	Copied from the incoming TradeCaptureReport.
48	SecurityID	Copied from the incoming TradeCaptureReport, if present.
55	Symbol	Copied from the incoming TradeCaptureReport, if present.
58	Text	If present, indicates reason for the message. Format is one letter reason code followed by colon and space followed by free form text message. Reason codes are: A = Admin D = Duplicate TradeReportID H = Halted Y = Symbol Not Supported Z = Unforseen Reason m = Market Access Risk Limit Exceeded o = Max Open Trade Count Exceeded
60	TransactTime	Copied from the incoming TradeCaptureReport.
75	TradeDate	Copied from the incoming TradeCaptureReport.
150	ExecType	Copied from the incoming TradeCaptureReport.
207	SecurityExchange	Copied from the incoming TradeCaptureReport, if present.
375	ContraBroker	BATS: Trade Reported on Cboe TRF
487	TradeReportTransType	Copied from the incoming TradeCaptureReport.
571	TradeReportID	Unique identifier for the trade report confirm as provided by Cboe
572	TradeReportRefID	Contains the TradeReportID of the original trade capture report to which this message relates
573	MatchStatus	Will be 0 (Matched) for confirm, and 1 (Unmatched) for a decline
574	MatchType	Copied from the incoming TradeCaptureReport.
828	TrdType	Omitted if TradePriceCondition (1839) = 14 (Trade with Price Improvement) is set. Otherwise, copied from the incoming TradeCaptureReport.
829	TrdSubType	Copied from the incoming TradeCaptureReport.
855	SecondaryTrdType	Copied from the incoming TradeCaptureReport.
856	TradeReportType	Will be 2 (Accept) for a confirm, 3 (Decline) for a decline and 0 (Submit) for an unsolicited change.
1003	TradeID	Id representing the trade, as seen on outbound market data. Allocated by Cboe as per the MiFID II definition of a Transaction Identification Code. Required to amend, cancel or release a report.
1123	TradeHandlingInstr	Copied from the incoming TradeCaptureReport.
1390	TradePublishIndicator	Will be 2 (Deferred Publication) if deferral is requested and the trade is eligible for such. Otherwise, copied from the incoming TRADECAPTUREREPORT.
1430	VenueType	Copied from the incoming TradeCaptureReport.
1838	NoTradePriceConditions	Copied from the incoming TradeCaptureReport, if present.

Repeating Group *TradePriceConditionGrp* must occur the number of times specified in *NoTradePriceConditions* (1838)			
	...	...	Entire block copied from incoming TradeCaptureReport, although order may be adjusted

Tag	Name	Description
2405	*ExecMethod*	Copied from the incoming `TradeCaptureReport`, if present.
2667	*AlgorithmicTradeIndicator*	Copied from the incoming `TradeCaptureReport`, if present.
7570	*RptTime*	Indicates the time at which a deferred trade report will be automatically published. Where *RptTime* falls outside of the systems operating time, the report will be published during operating hours on the next trading day. When no deferral is requested, or when the trade does not qualify for a deferral, any time returned will match *TransactTime* (60). Microsecond level resolution.
8013	*TrdRegPublication Reasons*	Waiver(s) or deferral(s) used for the trade. If more than one Pre-Trade Transparency Waiver or Deferral has been requested by the Participant and successfully applied by the system, the values will be seperated by a space (e.g. 4 5).
9882	*FeeCode*	Specific fee code associated with the trade. See the Fee Schedule for the respective market for possible values.
552	*NoSides*	Indicates the number of instances of the repeating group *TrdCapRpt-SideGrp* to follow.

Repeating Group *TrdCapRptSideGrp* must occur the number of times specified in *NoSides* (552)			
	...	...	Entire block copied from incoming TradeCaptureReport
	7772	*CentralCounterparty*	The CCP handling this trade leg for a confirm, and not present for a decline: NONE = No Clearing

	Standard Message Trailer	

5.3 Quote Status Report

A 'Quote Status Report' message is sent to the participant to acknowledge acceptance or otherwise of each 'Quote' or 'Quote Cancel' message.

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = `AI`
117	*QuoteID*	Copied from the corresponding `Quote` / `QuoteCancel`, if applicable. Unsolicited QuoteStatus messages can be sent due to system events, and in such circumstances, *QuoteID* may be omitted.
55	*Symbol*	Copied from the incoming `Quote/QuoteCancel`, if present.
48	*SecurityID*	Copied from the incoming `Quote/QuoteCancel`, if present.
22	*IDSource*	Copied from the incoming `Quote/QuoteCancel`, if present.
207	*SecurityExchange*	Copied from the incoming `Quote/QuoteCancel`, if present.
15	*Currency*	Copied from the incoming `Quote/QuoteCancel`, if present.

132	*BidPx*	For valid Quote messages (*QuoteStatus* (297) = 0 or 1), the new price and size fields are returned in the corresponding QuoteStatus message. These may be zero if there is no applicable quote on this side. Typically, these will match the values sent in the corresponding Quote, unless that Quote message opted to change one side of the quote only.
133	*OfferPx*	The quote's offer price, or zero. See *BidPx* (132) for more details.
134	*BidSize*	The quote's bid quantity, or zero. See *BidPx* (132) for more details.
135	*OfferSize*	The quote's offer quantity, or zero. See *BidPx* (132) for more details.
297	*QuoteStatus*	Indicates the acceptance or otherwise of a Quote or QuoteCancel message. 0: Accepted - indicates acceptance of a Quote message. 1: Cancelled - sent in respose to a successful QuoteCancel message. 5: Rejected - sent in respose to a Quote or QuoteCancel message, indicating the message could not be accepted.
58	*Text*	If present, indicates a reason the applicable Quote or QuoteCancel message could not be accepted.
	Standard Message Trailer	

6 Example Message Flow

6.1 New Trade Capture Reports

Below illustrates an example of key elements of the message flow for a new trade capture report.

Participant to Cboe:

- *MsgType* (35) = `AE` - Trade Capture Report
- *TradeReportID* (571) = `CLIENTID111` - Day-unique ID chosen by client
- *TradeReportTransType* (487) = 0 - New
- *TradeReportType* (856) = 0 - Submit

Cboe to Participant (Technical Ack) - if accepted:

- *MsgType* (35) = `AR`
- *TrdRptStatus* (939) = 0 - Accepted
- *TradeReportID* (571) = `CLIENTID111` - Day-unique ID chosen by client copied from the incoming trade capture report.
- *TradeReportRefID* (572) = `WXYZ1234` - Unique identifer for the trade capture report provided by Cboe.
- *TradeReportTransType* (487) = 0 - New
- *TradeReportType* (856) = 0 - Submit

Cboe to Participant (Business Ack)

- *MsgType* (35) = `AE`
- *TradeReportID* (571) = `WXYZ1234` - Unique identifier for the trade report confirm as provided by Cboe.
- *TradeReportRefID* (572) = `CLIENTID111` - Contains the TradeReportID of the original trade capture report to which this message relates.
- *TradeID* (1003) = `ABCD1234` - Represents the trade as seen on outbound market data allocated by Cboe. Requires to amend, cancel or release a report
- *TradeReportTransType* (487) = 0 - New
- *TradeReportType* (856) = 2 - Accept

TradeID (1003) = is a Cboe allocated ID as per the MiFID II definition of a Transaction Identification Code. This is the ID seen on outbound market data (Trade Message, Extended Trade Message and Trade Message Unknown).

6.2 Trade Capture Report Cancellations

Below illustrates an example of key elements of the message flow for cancellation of previously confirmed trade captures.

Participant to Cboe:

- *MsgType* (35) = `AE` - Trade Capture Report
- *TradeID* (1003) = `ABCD1234` - The TradeID for the confirmed trade report
- *TradeReportTransType* (487) = 1 - Cancel

Cboe to Participant (Technical Ack) - if rejected:

- *MsgType* (35) = `AR`
- *TrdRptStatus* (939) = 1 - Rejected
- *Text* (58) = `Reason for reject`

Cboe to Participant (Technical Ack) - if accepted:

- *MsgType* (35) = `AR`
- *TrdRptStatus* (939) = 0 - Accepted

Cboe to Participant - if cancellation is declined:

- *MsgType* (35) = `AE`
- *TradeReportTransType* (487) = 1 - Cancel
- *TradeReportType* (856) = 3 - Decline
- *Text* (58) = `Reason for decline`

Cboe to Participant - if cancellation is confirmed:

- *MsgType* (35) = `AE`
- *TradeReportTransType* (487) = 1 - Cancel
- *TradeReportType* (856) = 2 - Accept

6.3 Trade Capture Report Amendments

Below illustrates an example of key elements of the message flow for amendment of previously confirmed trade captures.

Participant to Cboe:

- *MsgType* (35) = `AE` - Trade Capture Report
- *TradeID* (1003) = `ABCD1234` - The TradeID for the confirmed trade report
- *TradeReportTransType* (487) = 2 - Replace

Cboe to Participant (Technical Ack) - if rejected:

- *MsgType* (35) = `AR`
- *TrdRptStatus* (939) = 1 - Rejected
- *Text* (58) = `Reason for reject`

Cboe to Participant (Technical Ack) - if accepted:

- *MsgType* (35) = `AR`
- *TrdRptStatus* (939) = 0 - Accepted

Cboe to Participant - if amendment is declined:

- *MsgType* (35) = `AE`
- *TradeReportTransType* (487) = 2 - Replace
- *TradeReportType* (856) = 3 - Decline
- *Text* (58) = `Reason for decline`

Cboe to Participant - if amendment is confirmed:

- *MsgType* (35) = `AE`
- *TradeReportTransType* (487) = 2 - Replace
- *TradeReportType* (856) = 2 - Accept

6.4 Deferred Publication Trade Reports

Customers reporting trades subject to deferred publication have the option of performing the deferment within their own systems, or asking Cboe to perform the deferment for them. To provide a consistent interface and to allow Cboe to distinguish between deferment eligible trades that have been delayed in the Participant's system and large, deferment-ineligible trades that have been reported late, Participants must flag trade reports appropriately. All trade reports subject to deferment must be reported with *TradePublishIndicator* (1390) = 2 to indicate that the trade is believed to be deferment eligible and should not be regarded as late (unless reported after the end of eligible deferment period).

Cboe will automatically hold onto the trade for the remainder of the eligible deferment period. To release the trade, follow the trade report with an immediate release (using *TradeReportTransType* (487) = 3).

Below illustrates an example of key elements of the message flow for deferred publication of trades captures.

Participant to Cboe:

- *MsgType* (35) = AE - Trade Capture Report
- *TradePublishIndicator* (1390) = 2 - Deferred Publication
- *TransactTime* (60) = Time of Trade - As long as this is within the acceptable deferment period, the trade will not be regarded as late

Cboe to Participant (Technical Ack) - if rejected:

- *MsgType* (35) = AR
- *TrdRptStatus* (939) = 1 - Rejected
- *Text* (58) = Reason for reject

Cboe to Participant (Technical Ack) - if accepted:

- *MsgType* (35) = AR
- *TrdRptStatus* (939) = 0 - Accepted

Cboe to Participant - if report is declined:

- *MsgType* (35) = AE
- *TradeReportTransType* (487) = 0 - New
- *TradeReportType* (856) = 3 - Decline
- *Text* (58) = Reason for decline

Cboe to Participant - if report is confirmed and deferment permitted:

- *MsgType* (35) = AE
- *TradeReportTransType* (487) = 0 - New
- *TradeReportType* (856) = 2 - Accept
- *TradePublishIndicator* (1390) = 2 - Deferred Publication

Cboe to Participant - if report is confirmed and deferment is not permitted:

- *MsgType* (35) = AE
- *TradeReportTransType* (487) = 0 - New
- *TradeReportType* (856) = 2 - Accept
- *TradePublishIndicator* (1390) = 1 - Publish trade (immediately)
- *Text* (58) = A: Trade accepted, but ineligible for deferment - Exact text may vary

---

Then, once the Participant would like the deferred trade released to the market (note - this may be immediately if the Participant has held onto the trade for the deferment period).

Participant to Cboe:

- *MsgType* (35) = `AE` - Trade Capture Report
- *TradeID* (1003) = `ABCD1234` - The TradeID for the confirmed trade report
- *TradeReportTransType* (487) = 3 - Release

Cboe to Participant (Technical Ack) - if rejected:

- *MsgType* (35) = `AR`
- *TrdRptStatus* (939) = 1 - Rejected
- *Text* (58) = `Reason for reject`

Cboe to Participant (Technical Ack) - if accepted:

- *MsgType* (35) = `AR`
- *TrdRptStatus* (939) = 0 - Accepted

Cboe to Participant - if release is declined:

- *MsgType* (35) = `AE`
- *TradeReportTransType* (487) = 3 - Release
- *TradeReportType* (856) = 3 - Decline
- *Text* (58) = `Reason for decline`

Cboe to Participant - if release is confirmed:

- *MsgType* (35) = `AE`
- *TradeReportTransType* (487) = 3 - Release
- *TradeReportType* (856) = 2 - Accept

7 Common Session Level Issues

Cboe uses FIX 4.2 or 4.4 as specified by the applicable FPL Documents (Version 4.2 with Errata 20010501 or Version 4.4 with Errata 20030618) with business level extensions as described in this document. The session level of the FPL specification is followed as closely as possible.

The errata for version 4.2 cleared up many session level ambiguities present in the earlier version 4.2 (March 1, 2000). The following sections emphasize a few common problem areas in implementations of the FIX session protocol.

Typographical conventions:

- Anchor locations in the FPL document are shown in blue.

- Text in **bold** was emphasized in the original FPL specification.

- Emphasis added by Cboe is shown in purple.

- Notes added by Cboe are shown in green.

7.1 Ordered Message Processing

From Financial Information Exchange Protocol/FIX Message Format and Delivery/Ordered Message Processing:

> The FIX protocol assumes complete ordered delivery of messages between parties. Implementers should consider this when designing message gap fill processes. Two options exist for dealing with gaps, either request all messages subsequent to the last message received or ask for the specific message missed while maintaining an ordered list of all newer messages. For example, if the receiver misses the second of five messages, the application could ignore messages 3 through 5 and generate a resend request for messages 2 through 5, **or**, preferably 2 through 0 (where 0 represents infinity). Another option would involve saving messages 3 through 5 and resending only message 2. In both cases, messages 3 through 5 should not be processing before message 2.

7.2 Logon

From Financial Information Exchange Protocol/Session Protocol/Logon:

> After the initiator has been authenticated, the acceptor will respond immediately with a confirming *Logon* message.

7.3 Message Recovery

From Financial Information Exchange Protocol/Session Protocol/Message Recovery:

> When the incoming sequence number does not match the expected number, corrective processing is required. Note that the SeqReset-**Reset** message ([Cboe: this refers only to *GapFillFlag* (123) = N] used only to recover from a disaster scenario vs. normal resent request processing) is an exception to this rule as it should be processed without regards to its *MsgSeqNum* (34). **If the incoming message has a sequence number less than expected and the PossDupFlag (43) is not set, it indicates a serious error. It is strongly recommended that the session be terminated and manual intervention be initiated**. If the incoming sequence number is greater than expected, it indicates that messages were missed and retransmission of the messages is requested via the *Resend Request* (see earlier section, *Ordered Message Processing*).
>
> . . .

If there are consecutive administrative messages to be resent, it is suggested that only one *SeqReset-GapFill* message be sent in their place. The sequence number of the *SeqReset-GapFill* message is the next expected outbound sequence number. The *NewSeqNo* (36) field of the GapFill message contains the sequence number of the highest administrative message in the group plus 1. For example, during a Resend operation there are 7 sequential administrative messages waiting to be resent. They start with sequence number 9 and end with sequence number 15. Instead of transmitting 7 GapFill messages (which is perfectly legal, but not network friendly), a *SeqReset-GapFill* message may be sent. **The sequence number of the Gap Fill message is set to 9 because the remote side is expecting that as the next sequence number.** The *NewSeqNo* (36) field of the Gap Fill message contains the number 16, because that will be the sequence number of the next message to be transmitted.

Sequence number checking is a vital part of FIX session management. However, a discrepancy in the sequence number stream is handled differently for certain classes of FIX messages. The table below lists the actions to be taken when the incoming sequence number is greater than the expected incoming sequence number.

> **NOTE: In all cases except the Sequence Reset – Reset message, the FIX session should be terminated if the incoming sequence number is less than expected and the PossDupFlag (43) is not set. A Logout message with some descriptive text should be sent to the other side before closing the session.**

Response by Message Type

Message Type	Action to Be Taken on Sequence # Mismatch
Logon	Must always be the first message transmitted. Authenticate and accept the connection. After sending a *Logon* confirmation back, send a *ResendRequest* if a message gap was detected in the *Logon* sequence number.

. . .

7.4 Resend Request

From Financial Information Exchange Protocol/Administrative Messages/Resend Request:

Note: the sending application may wish to consider the message type when resending messages; e.g., if a new order is in the resend series and a significant time period has elapsed since its original inception, the sender may not wish to retransmit the order given the potential for changed market conditions. (The Sequence Reset-Gap Fill message is used to skip message that a sender does not wish to resend.)

7.5 Sequence Reset – Gap Fill

From Financial Information Exchange Protocol/Administrative Messages/Sequence Reset (Gap Fill):

The sequence reset message is used by the sending application to reset the incoming sequence number on the opposing side. This message has two modes: "Sequence Reset – Gap Fill when *GapFillFlag* (123) is 'Y' and "Sequence Reset – Reset" when *GapFillFlag* (123) is 'N' or not present. The "Sequence Reset – Reset" mode should **only** be used to recover from a disaster situation which cannot e otherwise recovered via "Gap Fill" mode. The sequence reset message can be used in the following situations:

- During normal resend processing, the sending application may choose not to send a message (e.g., an aged order). The Sequence Reset – Gap Fill is used to mark the place of that message.

- During normal resend processing, a number of administrative messages are not resent, the Sequence Reset – Gap Fill message is used to fill the sequence gap created.

. . .

The sending application will initiate the sequence reset. **The message in all situations specifies the NewSeqNo (36) to reset as the value of the <u>next</u> sequence number immediately following the messages and/or sequence numbers being skipped.**

. . .

If the *GapFillFlag* (123) field is present (and equal to 'Y'), the *MsgSeqNum* (34) should conform to standard message sequencing rules (i.e., the *MsgSeqNum* (34) of the SeqReset-GapFill message should represent the beginning *MsgSeqNum* (34) in the gap fill range because the remote side is expecting that next message).

The sequence reset can only increase the sequence number. If a sequence reset is received attempting to decrease the next expected sequence number, the message should be rejected and treated as a serious error. It is possible to have multiple resend requests issued in a row (i.e., 5 to 10 followed by 5 to 11). If sequence number 8, 10, and 11 represent application messages while 5–7 and 9 represent administrative messages, the series of messages as a result of the resend request may appear as SeqReset-GapFill with *NewSeqNo* (36) of 8, message 8, SeqReset-GapFill with *NewSeqNo* (36) of 10, and message 10. This could then be followed by SeqReset-GapFill with *NewSeqNo* (36) of 8, message 8, SeqReset-GapFill with *NewSeqNo* (36) of 10, message 10, and message 11. One must be careful to ignore the duplicate SeqReset-GapFill which is attempting to lower the next expected sequence number. This can be detected by checking to see if its *MsgSeqNum* (34) is less than expected. If so, the SeqReset-GapFill is a duplicate and should be discarded.

8 Support

Please email questions or comments regarding this specification to tradedeskeurope@cboe.com.

Revision History

3rd June 2013	Version 1.0
	First non-draft release.
17th June 2013	Version 1.1
	QuoteID is mandatory on Quote and QuoteCancel messages. *Symbol* is expected after *NoQuoteEntries*. Tweak to *TradeReportType* (856) values returned in Trade Capture Reports confirmed or declined. Added message flow section.
18th June 2013	Version 1.2
	Refined some values under New/Cancel/Amend/Release scenario when confirming or declining to make more FIX compliant.
25th June 2013	Version 1.3
	Clarified values for *TradePublishIndicator* on Trade Capture Confirmations.
28th June 2013	Version 1.4
	Clarified example when requesting deferment of the publication of a trade that is not eligible for such. Added *CentralCounterparty* to Trade Capture Confirmation.
16th July 2013	Version 1.5
	Updated values used in example messages to match current requirements. Tweak to *TradeReportType* (856) for unsolicited changes. Allow the use of *PreviouslyReported* (570) to ease migration.
20th August 2013	Version 1.6
	Clarified behaviour of *TradeReportID* (571) for Participant submitted trade reports. Clarified expected messaging for Deferred publication trade reports. Clarified unknown symbol handling. Added *RptTime* (7570). Allow price and size fields to be omitted from Quote messages without cancelling the quote.
4th October 2013	Version 1.7
	Updated symbology section to include ISIN/currency lookup, minor currency and preferred symbol handling.
29th October 2013	Version 1.8
	Removed Broker Crossing Network suggestions.
21st March 2014	Version 1.9
	Addition of *ExecMethod* (2405) for trade captures.
25th March 2014	Version 1.10
	Clarification of *SecurityExchange* (207) usage for trade reports.
28th March 2014	Version 1.11
	Removed 'effective from' labels.
12th April 2014	Version 1.12
	Updated details on requirements for *QuoteID* (117).
20th May 2014	Version 1.13
	Bug fix location of *SendingTime* (52) (should be in session header). Use latest proposed name for MMT 3.7.
16 June 2014	Version 1.14
	Increased the number of decimal places supported for Trade Capture Reports.

25 June 2014	Version 1.15 Added changes to Trade Capture messages effective with the Cboe Q3 2014 release, being those that follow. *PublicationMode* (1390) becomes mandatory. Deprecated use of *VenueType* (1430) = X in favour of *VenueType* (1430) = 0. Deprecated use of *PreviouslyReported* (570) in favour of values in *PublicationMode* (1390). Added support for *PublicationMode* (1390) = 0 (Do Not Publish). Deprecated use of *TrdType* (828) = 59 in favour of *TrdType* (828) = 62. Deprecated use of *TrdType* (828) = 60 in favour of *TrdType* (828) = 63. Deprecated use of *SecondaryTrdType* (855) = 58 in favour of *SecondaryTrdType* (855) = 64. Moved *TradingSessionSubID* (625) into the repeating group. Added support for *OrderCategory* (1115). Added support for *TradingSessionSubID* (625) = 8 (Unspecified Auction).
25th July 2014	Version 1.16 Removed 'effective from 25th of July' labels.
30th July 2014	Version 1.17 Removed deprecated content.
10th November 2014	Version 1.18 Added new reject reason code.
19 January 2015	Version 1.19 Clarified usage of *OrderCategory* (1115). Added additional fields to TRADE CAPTURE REPORT ACK and Cboe to Participant TRADE CAPTURE REPORT. Added support for *GrossTradeAmt* (381). Changed *TradeHandlingInstr* (1123) in Cboe to Participant TRADE CAPTURE REPORT to echo back supplied value.
27 April 2015	Version 1.20 Remove 'Effective' notes now we're past their go live date
1 December 2015	Version 1.21 Clarified that *TransactTime* (60) is only optional when *TradeReportTransType* (487) = 0.
19 February 2016	Version 1.22 Updated for new branding.
1 February 2017	Version 1.23 Change in spec to support MMT v3.
8 February 2017	Version 1.24 Updated with review feedback for MMT v3.
20 March 2017	Version 1.25 Clarified TrdType values in TradeCaptureReport.
24 May 2017	Version 1.26 Corrected description for *TradePriceCondition* (1839) to not drive MMT v3 'NPFT' on market data
1 June 2017	Version 1.27 Removed *OnBehalfOfSubID* (116) and *DeliverToSubID* (129) from the message header as these are not currently supported on TCRs.
5 July 2017	Version 1.28 Removed *OnBehalfOfCompID* (115) from the message header as it is not currently supported on TCRs. Also removed the Service Bureau section.
19 July 2017	Version 1.29 MMT v3.04 support for Q4 2017 release.
28 September 2018	Version 1.30 Cboe will enforce port level day-uniqueness for *TradeReportID* (571).
15 November 2018	Version 1.31 Update *TradeReportID* (571) description.

28 February 2019	Version 1.32
	Clarified the bank code used to submit a trade report determines which APA (UK or EU) is being reported to, rather than the Cboe RIC suffix.