Cboe Titanium Cboe Global Cloud Feed Specification

Version 1.1.29

July 31, 2025

Contents

Introduction

Cboe Titanium Cboe Global Cloud allows customers to receive real-time data from Cboe Global Markets through an Apache Kafka service hosted in Amazon Web Services (AWS). Cboe offers multiple data feeds in the Cboe Global Cloud service and enables Cboe data availability and dissemination in multiple locations across the world. Cboe offers various secure connectivity and access options through the Cboe Global Cloud service. This document describes the technical specifications associated with all Cboe data products available within Cboe Global Cloud. Each data feed is described in a separate chapter with example messages provided to assist in development efforts.

Connecting to Cboe Global Cloud

Detailed instructions for using each of the following connectivity options can be found in the Cboe Global Cloud Setup Guide.

Connectivity Methods

Configuration information on each of the following methods will be provided during the onboarding process once an access method is selected.

PrivateLink (for connectivity from AWS accounts)

This option is a suitable access method if customer systems and/or processes are running within AWS Cloud (i.e., a current AWS customer).

Internet Gateway

This option is a suitable access method for customer systems and/or processes running on-premises in a local data center, or in another infrastructure provider's environment.

VPN

This option is a suitable access method for customer systems and/or processes running on-premises in a local data center, or in another infrastructure provider's environment, where a virtual private connection is preferred over internet delivery.

Data Feed Kafka Topics (by Region)

Region: Hong Kong (ap-east-1)

Table 1. Hong Kong (ap-east-1)

NAME	TOPIC NAME
Cboe One Summary	cos-b1s-ape1
Cboe One Premium	cop-b1p-ape1
BZX TOP	bzx-tbc-ape1
EDGX TOP	edgx-tbc-ape1
CFE TOP	cfe-mtop-ape1
Cboe Global Indices	cccy-idx- ape1 cgi-idx-ape1 csmi-idx-ape1 definition-idx-ape1 ftse-idx- ape1 inav-idx- ape1 msci-idx- ape1 mstarc-idx-ape1 mstar-idx-ape1

Region: London (eu-west-2)

Table 2. London (eu-west-2)

NAME	TOPIC NAME
Cboe One Summary	cos-b1s- euw2
Cboe One Premium	cop-b1p- euw2
BZX TOP	bzx-tbc- euw2
EDGX TOP	edgx-tbc-euw2
CFE TOP	cfe-mtop-euw2
Cboe Global Indices	cccy-idx- euw2 cgi-idx-euw2 csmi-idx-euw2 definition-idx-euw2 ftse-idx- euw2 inav-idx- euw2 msci-idx- euw2 mstarc-idx-euw2 mstar-idx-euw2
BXE Top	bxe-tbc-euw2
BXE Last Sale	bxe-ls-euw2
CXE Top	cxe-tbc-euw2
CXE Last Sale	cxe-ls-euw2
DXE Top	dxe-tbc-euw2
DXE Last Sale	dxe-ls-euw2

Region: Virginia (us-east-1)

Table 3. Virginia (us-east-1)

NAME	TOPIC NAME
Cboe One US Summary	cos-b1s-use1
Cboe One US Premium	cop-b1p- use1
Cboe One Canada Summary	cop-b1s-ca-use1
Cboe One Canada Premium	cop-b1p-ca-use1
BZX TOP	bzx-tbc- use1
EDGX TOP	edgx-tbc- use1
CFE TOP	cfe-mtop-use1
Cboe Global Indices	cccy-idx- use1 cgi-idx-use1 csmi-idx-use1 definition-idx-use1 ftse-idx- use1 inav-idx- use1 msci-idx- use1 mstarc-idx-use1 mstar-idx-use1

Region: Canada (ca-central-1)

Table 4. Canada (ca-central-1)

NAME	TOPIC NAME
Cboe One Canada Summary	cos-b1s-ca-cac1
Cboe One Canada Premium	cop-b1p-ca-cac1

Region: Singapore (ap-southeast-1)

Table 5. Singapore (ap-southeast-1)

NAME	TOPIC NAME
Cboe One Summary	cos-b1s-aspe1
Cboe One Premium	cop-b1p-apse1
BZX TOP	bzx-tbc-apse1
EDGX TOP	edgx-tbc-apse1
CFE TOP	cfe-mtop-apse1
Cboe Global Indices	cccy-idx- apse1 cgi-idx-apse1 csmi-idx-apse1 definition-idx-apse1 ftse-idx- aspe1 inav-idx- apse1 msci-idx- apse1 mstar-idx-apse1

Region: Sydney (ap-southeast-2)

Table 6. Sydney (ap-southeast-2)

NAME	TOPIC NAME
Cboe One Summary	cos-b1s-apse2
CXA TOP	cxa-mtop-apse2
CXA PITCH	cxa-bpitch-apse2
BXE TOP	bxe-tbc-apse2
CXE TOP	cxe-tbc-apse2
DXE TOP	dxe-tbc-apse2
EDGX TOP	edgx-tbc-apse2

Region: Tokyo (ap-northeast-1)

Table 7. Tokyo (ap-northeast-1)

NAME	TOPIC NAME
CXJA PITCH	cxja-bpitch-apne1
CXJS PITCH	cxjs-bpitch-apne1

Recommended Bandwidth

Bandwidth recommendations for each feed are outlined below. These recommendations are reflective of a consumer process reading messages from a topic as quickly as the producer publishes them.

Table 8. Recommended Bandwidth

NAME	BANDWIDTH
Cboe One Premium	350 Mb
Cboe One Summary	150 Mb
Cboe One Canada Premium	70 Mb
Cboe One Canada Summary	20 Mb
BZX TOP	65 Mb
EDGX TOP	60 Mb
CXA PITCH	10 Mb
CXJA PITCH	10 Mb
CXJS PITCH	5 Mb
CXA TOP	5 Mb
CFE TOP	1 Mb
Cboe Global Indices	1 Mb
BXE Top	1 Mb
BXE Last Sale	
CXE Top	
CXE Last Sale	
DXE Top	
DXE Last Sale	

Protocol

Cboe Global Cloud users may connect to any of the Cboe Global Cloud regional clusters to receive the available Cboe data feeds.

Cboe data feeds via Cboe Global Cloud cannot be used to enter orders. For order entry relating to non-cloud exchange connectivity, refer to the specifications below:

- Cboe US Equities FIX and BOE
- Cboe EU Equities FIX and BOE
- Cboe AU Equities FIX and BOE
- Cboe JP Equities FIX and BOE
- Cboe Canadian Equities FIX and BOE

Kafka Configuration

Cboe does not recommend any specific Apache Kafka distribution or managed service, but general documentation can be found within https://kafka.apache.org/documentation/. Additionally, information related to connecting to ASW MSC can be found within https://docs.aws.amazon.com/msk/latest/developerguide/produce-consume.html.

Authentication

Authentication to the Kafka cluster will utilize mutual TLS. During the onboarding process, an authentication certificate will be provided.

Broker Information

Broker information for each subscription to be provided during the onboarding process.

The number of brokers for each feed provided is outlined below:

Table 9. Broker Information

NAME	NUMBER OF BROKERS	PARTITIONS PER TOPIC
Cboe One Summary	6	36
Cboe One Premium	6	36
BZX TOP	6	35
EDGX TOP	6	32
CXA TOP	3	2
CXA PITCH	3	2
CXJA PITCH	3	4
CXJS PITCH	3	4
CFE TOP	3	2
Cboe Global Indices	3	3
BXE Top	3	12
BXE Last Sale	3	12
CXE Top	3	12
CXE Last Sale	3	12
DXE Top	3	12
DXE Last Sale	3	12

Cboe One Feed

The Cboe One Feed delivers consolidated quote, trade, and Aggregated Depth At Price (ADAP) information for all Cboe US Equities (Cboe One US) and MATCHNow, NEO-L, NEO-N, and NEO-D (Cboe One Canada) books via connection to Cboe's Amazon MSK cluster using the Cboe One Feed protocol. The feed consists of **`Clear Quote`**, **`Symbol Summary`**, **`Best Quote Update`**, **`Market Status`**, **`ADAP`**, **`RPI`**, **`Trade`**, **`Trade Break`**, **`Trading Status`**, **`Opening/Closing Price`**, and **`End of Day Summary`** messages.

The feed is sourced from the Cboe Amazon MSK Kafka cluster. The Cboe US Equities topics have 36 partitions, while the Cboe One Canada topics have 4 partitions. Cboe reserves the right to add partitions to these topics for performance reasons if it becomes necessary.

It is important to note that the Cboe One Feed is an aggregated feed. Updates for a symbol are sent as capacity for the feed allows. The image for a symbol will be current at the time of delivery, but multiple updates may be combined into a single update. The interval between updates is dependent on the market conditions and the capacity configuration of the particular Cboe One feed.

The Cboe One Feed is available with different combinations of ADAP levels and update rates to meet the needs of our members.

Table 10. Current Cboe One Feed Descriptions

NAME	BEST QUOTE	ADAP LEVELS
Cboe One US Summary	Yes	0
Cboe One US Premium	Yes	5
Cboe One Canada Summary	Yes	0
Cboe One Canada Premium	Yes	5

Message Format

The messages that make up the Cboe One protocol are formatted with JSON, and each message contains the *Sequence* field (see Sequence Field on page 23), which handles sequencing and delivery integrity.

The Cboe One data feed is comprised of a series of dynamic length sequenced messages. Each message begins with the *messageType (m)* field. **Cboe reserves the right to add message types and grow the length** of any message without notice. Cboe One users should develop their decoders to deal with unknown message types and messages that grow beyond the expected length. Messages will only be grown to add additional data to the end of a message.

Data Types

The following field types are used within Cboe One feed. All types are rendered in printable ASCII.

- **Alphanumeric** fields are strings of characters that are not meant for conversion into another data type.
- **Integer** fields are convertible into unsigned or long unsigned integers.
- **Price** fields are convertible into floats or doubles.
- **Bit Field** fields are fixed width fields with each bit representing a Boolean flag (the 0 bit is the lowest significant bit; the 7 bit is the highest significant bit).

Sequence Field

The *Sequence* field is used for all Cboe One messages delivered via Amazon MSK. Examples can be found in the messages below. Note that when the Cboe One feed is spread across multiple partitions, messages may not always be received in sequence order.

Table 11. Sequence Field

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Sequence	*s*	Integer	Sequence representing the order that the messages were received by the Kafka producer.

Cboe One Update Messages

Clear Quote Message Fields

The **`Clear Quote`** message instructs feed recipients to clear all quotes and Summary and/or ADAP information for the specified symbol on the specified market(s). This message does not affect the executed volume of the symbol.

Table 12. Clear Quote Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*m*	Alphanumeric	CQ
timestamp	*ts*	Integer	Timestamp of the last matching engine message that updated the quote information for this symbol in the Cboe One server's cache. Encoded as the number of nanoseconds since midnight.
Symbol	*sy*	Alphanumeric	Relevant symbol.
marketCenter	*mc*	Alphanumeric	* = All Cboe Markets Y = BYX (US only) Z = BZX (US only) A = EDGA (US only) X = EDGX (US only) M = MATCHNow (Canada only) L = NEO-L (Canada only) N = NEO-N (Canada only) D = NEO-D (Canada only) r = NEO-Cross (Canada only) t = NEO-SST (Canada only)

Symbol Summary Message Fields

The **`Symbol Summary`** message delivers the Cboe consolidated best bid/offer and total executed volume across all 8 (eight Cboe One US and Cboe One Canada) equities books.

Table 13. Symbol Summary Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*m*	Alphanumeric	S
timestamp	*ts*	Integer	Timestamp of the last matching engine message that updated the quote information for this symbol in the Cboe One server's cache. Encoded as the number of nanoseconds since midnight.
symbol	*sy*	Alphanumeric	Relevant symbol.
cumulativeVolume	*cv*	Integer	Cumulative number of shares traded today across all 8 Cboe books.
bestBidPrice	*Bb*	Price	Cboe Consolidated best bid price.
bestBidSize	*Bs*	Integer	Cboe Consolidated number of buy-side shares available for this symbol.
bestAskPrice	*Ba*	Price	Cboe Consolidated best ask price.
bestAskSize	*As*	Integer	Cboe Consolidated number of sell-side shares available for this symbol.
nationalVolume	*sv*	Integer	In the US, cumulative number of shares traded today as reported to the CTA and UTP SIPs. In Canada, the sum of two values: the current Cboe Cumulative Executed Volume and the 15-minute delayed executed volume of other Canadian exchanges as reported by TMX IP.
flags	*f*	Bit Field	**Bit 0: SIP Volume Status** 0 : SIP volume data is complete. 1 : SIP volume data may not be complete due to an unrecoverable gap on the incoming feed **Bits 1-7: Reserved**

Best Quote Update Message Fields

The **`Best Quote Update`** message is used to update one side of the Cboe consolidated quote information for a symbol. Since the message only updates one side of the quote the previous value for the other side of the quote remains in effect.

This message does not affect the executed volume of the symbol.

Table 14. Best Quote Update Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*m*	Alphanumeric	Q
timestamp	*ts*	Integer	Timestamp of the last matching engine message that updated the quote information for this symbol in the Cboe One server's cache. Encoded as the number of nanoseconds since midnight.
symbol	*sy*	Alphanumeric	Relevant symbol.
side	*sd*	Alphanumeric	B = Buy Side S = Sell Side
bestPrice	*P*	Price	Cboe Consolidated best price.
bestSize	*S*	Integer	Cboe Consolidated number of shares available for this symbol.

Cboe Market Status Message Fields

The **`Cboe Market Status`** message is disseminated to reflect a change in the status of a market. All markets should be assumed to be Normal unless otherwise indicated by a **`Cboe Market Status`** message.

The Incomplete market status is used to indicate the feed has not delivered updates for all a market center's symbols and that the feed is transitioning to Normal. If a market center transitions from Normal to Excluded the feed will deliver updates of symbol quote/ADAP information to properly reflect the state of the combined book. At the start of a market center's transition from Excluded to Normal a **`Cboe Market Status`** message will be sent with Incomplete for the market center's status. Symbol quote/ADAP information will then be sent for all applicable symbols. Once the market center's symbol information has been disseminated a **`Cboe Market Status`** message will be delivered with a Normal market status.

Table 15. Cboe Market Status Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*m*	Alphanumeric	MS
timestamp	*ts*	Integer	Timestamp of when the Market Status changed for the specified Market Center. Encoded as the number of nanoseconds since midnight.
marketCenter	*mc*	Alphanumeric	Y = BYX (US only) Z = BZX(US only) A = EDGA(US only) X = EDGX(US only) M = MATCHNow (Canada only) L = NEO-L(Canada only) N = NEO-N(Canada only) D = NEO-D(Canada only) r = NEO-Cross (Canada only) t = NEO-SST (Canada only)
marketStatus	*ms*	Alphanumeric	N = Normal E = Excluded from Symbol Summary and ADAP updates I = Incomplete
sessionIndicator	*sn*	Alphanumeric	R = Regular trading session P = Pre- or post-market session

ADAP Message Fields

Each **ADAP** message delivers one or more updates for a Symbol's ADAP book. Each **ADAP** message contains one or more ADAP Blocks. A receiving process should interpret each ADAP Block as a replacement for any previously delivered ADAP Blocks at that price level.

A quantity of 0 indicates that the price level is either no longer available or the price level is not within the number ADAP levels maintained by the feed. In either case a receiving process should delete a price level with a 0 quantity from its cache.

Table 16. ADAP Message Fields

FIELD	JSON	VALUE/TYPE	DESCRIPTION
messageType	*m*	Alphanumeric	A
timestamp	*ts*	Integer	Timestamp of the last matching engine message that updated the quote information for this symbol in the Cboe One server's cache. Encoded as the number of nanoseconds since midnight.
symbol	*sy*	Alphanumeric	Relevant symbol.
flags	*f*	Bit Field	**Bit 0: Clear ADAP** 0 = Keep any ADAP information for this symbol. 1 = Delete any ADAP information for this symbol before applying ADAP Blocks. **Bit 1: ADAP Complete** 0 = The ADAP view for this symbol is complete. 1 = More ADAP updates for this symbol to follow in another ADAP message. **Bit 2: Short/Long Block(s)** 0 = Short Update ADAP Block(s) to follow 1 = Long Update ADAP Block(s) to follow **Bit 3-7: Spare**
adapBlocks	*ab*	JSON List	List of ADAP Block JSON objects, defined below.

Table 17. ADAP Block Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
marketCenter	*mc*	Alphanumeric	Y = BYX (US only) Z = BZX (US only) A = EDGA (US only) X = EDGX (US only) L = NEO-L (Canada only) N = NEO-N (Canada only) r = NEO-Cross (Canada only) t = NEO-SST (Canada only)
side	*Sd*	Alphanumeric	B = Buy Side S = Sell Side
price	*P*	Price	*Price* level to add/update for Market Center's ADAP book.

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
size	*Sz*	Integer	*Quantity* of shares at this price level in the Market Center's ADAP book. A value of zero implies deletion of this ADAP level.

Retail Price Improvement (RPI) Message Fields

The **`Retail Price Improvement (RPI)`** message is a retail liquidity indicator that includes symbol and side, but not price and size. An **`RPI`** message will be disseminated when there is a retail price improving order present for a symbol on any Cboe Exchange order book OR to indicate a RPI order is no longer available. RPI orders offer price improvement in increments of $.001 to Retail Member Organizations.

Table 18. Retail Price Improvement (RPI) Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*M*	Alphanumeric	RP
timestamp	*Ts*	Integer	Timestamp of the matching engine RPI message emitted by the specified Market Center. Encoded as the number of nanoseconds since midnight.
symbol	*sy*	Alphanumeric	Relevant symbol.
marketCenter	*mc*	Alphanumeric	Y = BYX Z = BZX A = EDGA X = EDGX
rpiIndicator	*rn*	Alphanumeric	B = Buy Side RPI S = Sell Side RPI A = Buy & Sell RPI N = No RPI

Trade Message Fields

`Trade` messages are sent when an order is executed in whole or in part on a Cboe exchange. The last-sale eligible status (*Flags* field Bit 1) is derived based on four criteria:

1. The *Session Indicator* (see Cboe Market Status Message Fields on page 27) must be in the regular session.
2. The *Last Quantity* must be at least one round lot.
3. The *Market Status* (see Cboe Market Status Message Fields on page 27) for the executing exchange must be Normal.
4. The *Transaction Time* of the event must be within 10 seconds of the current time.

Table 19. Trade Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*m*	Alphanumeric	T
timestamp	*ts*	Integer	The time the trade occurred on the specified Market Center. Encoded as the number of nanoseconds since midnight.
symbol	*sy*	Alphanumeric	Relevant symbol.
marketCenter	*mc*	Alphanumeric	Market Center on which the last trade was executed: Y = BYX (US only) Z = BZX (US only) A = EDGA (US only) X = EDGX (US only) M = MATCHNow (Canada only) L = NEO-L (Canada only) N = NEO-N (Canada only) D = NEO-D (Canada only) r = NEO-Cross (Canada only) t = NEO-SST (Canada only)
executionId	*e*	Alphanumeric	Market center specific execution identifier of this Execution. *Execution ID* is also referenced in the Trade Break message.
lastPrice	*lp*	Price	Last trade price.
lastSize	*ls*	Integer	Last trade quantity.
cumulativeVolume	*cv*	Integer	Cumulative number of shares traded today across all 8 Cboe books.
nationalVolume	*sv*	Integer	In the US, cumulative number of shares traded today as reported to the CTA and UTP SIPs. In Canada, the sum of two values: the current Cboe Cumulative Executed Volume and the 15-minute delayed executed volume of other Canadian exchanges as reported by TMX IP.
flags	*f*	Bit Field	Bit 0: SIP Volume Status 0 = SIP volume data is complete.

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
			1 = SIP volume data may not be complete due to an unrecoverable gap on the incoming feed. Bit 1: Last Sale Eligible 0 = Trade is not last-sale eligible 1 = Trade is last-sale eligible Bit 2-7: Reserved

Trade Break Message Fields

The **`Trade Break`** message is sent whenever an execution on a Cboe exchange is broken. Trade breaks are rare and only affect applications that rely upon Cboe execution based data.

Table 20. Trade Break Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*M*	Alphanumeric	TB
timestamp	*Ts*	Integer	The time the trade break occurred on the specified Market Center. Encoded as the number of nanoseconds since midnight.
symbol	*Sy*	Alphanumeric	Relevant symbol.
marketCenter	*Mc*	Alphanumeric	Market Center on which the last trade was executed: Y = BYX (US only) Z = BZX (US only) A = EDGA (US only) X = EDGX (US only) M = MATCHNow (Canada only) L = NEO-L (Canada only) N = NEO-N (Canada only) D = NEO-D (Canada only) r = NEO-Cross (Canada only) t = NEO-SST (Canada only)
executionId	*E*	Alphanumeric	Market center specific execution identifier of trade to be broken.
cumulativeVolume	*Cv*	Integer	Cumulative number of shares traded today across all 8 Cboe books.
nationalVolume	*Sv*	Integer	In the US, cumulative number of shares traded today as reported to the CTA and UTP SIPs. In Canada, the sum of two values: the current Cboe Cumulative Executed Volume and the 15-minute delayed executed volume of other Canadian exchanges as reported by TMX IP.
flags	*F*	Bit Field	**Bit 0: SIP Volume Status** 0 = SIP volume data is complete. 1 = SIP volume data may not be complete due to an unrecoverable gap on the incoming feed. **Bits 1-7: Reserved**

Trading Status Message Fields

The **`Trading Status`** message is used to indicate the current trading status of a security on a Cboe exchange. A **`Trading Status`** message will be sent whenever a security's trading status changes.

A **`Trading Status`** message will be sent:

- for Regulatory Halts in any security as well as the Trading resumption for the same security.
- for Cboe Listed securities that are in a Quoting period for auctions.
- to indicate a Reg SHO price test is in effect.

Table 21. Trading Status Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*m*	Alphanumeric	TS
timestamp	*ts*	Integer	Timestamp of the matching engine trading status message emitted by the specified Market Center. Encoded as the number of nanoseconds since midnight.
symbol	*sy*	Alphanumeric	Relevant symbol.
marketCenter	*mc*	Alphanumeric	Y = BYX (US only) Z = BZX (US only) A = EDGA (US only) X = EDGX (US only) M= MATCHNow (Canada only) L = NEO-L (Canada only) N = NEO-N (Canada only) D = NEO-D (Canada only) r = NEO-Cross (Canada only) t = NEO-SST (Canada only)
tradingStatus	*h*	Alphanumeric	A = Accepting Orders for Queuing F = Halted (full) (NEO only) (effective 08/11/25) H = Halted M = Delayed Closing (NEO only) (effective 08/11/25) O = Pre-close (NEO only) (effective 08/11/25) P = Post-close (NEO only) (effective 08/11/25) Q = Quote-Only S = Exchange Specific Suspension T = Trading X = Extended Trading (NEO only) (effective 08/11/25)
regShoAction	*rs*	Alphanumeric	0 = No price test in effect 1 = Reg SHO price test restriction in effect

Opening/Closing Price Message Fields

The **`Opening/Closing Price`** message is used to indicate the Opening or Closing price of a security on any of the Cboe exchanges and the primary listing market. An **`Opening/Closing Price`** message will be sent whenever the opening or closing price of a security is established. The opening or closing price is established once across all Cboe exchanges (not per-exchange). Additionally, an **`Opening/Closing Price`** message will be sent after it is received from the CTA or UTP SIP (in US markets) or from the TMX IP feeds (in Canadian markets) once the delay period has expired, currently 15 minutes.

- For Cboe listed securities, the opening and closing prices will be sent when they are received from the listing market and again when received from the SIP and the delay period has expired.
- For non-Cboe listed securities, the opening price of a security is defined as the first eligible trade received that occurred on or after 9:30:00 from any Cboe exchange. If no eligible trade is received by 9:35:00, no opening price will be reported from a Cboe exchange. A message will also be sent after the official open from the listing exchange is received and the SIP delay period has expired with *Market Center* set to `C` or `U`.
- For non-Cboe listed securities, the closing price of a security is the last eligible trade received when any Cboe exchange timestamp exceeds the end of the regular session (normally 16:00:00). If no eligible trade has occurred prior to the close, no closing price will be reported from a Cboe exchange. A message will also be sent after the official close from the listing exchange is received and the SIP delay period has expired with *Market Center* set to `C` or `U`.

Cboe One Canada:

- For NEO listed securities, the opening and closing prices will be sent when they are received from the listing market.
- For non-NEO listed securities, the opening price of a security is defined as the first round lot trade received from the security's listing exchange that occurred on or after the listing exchange's opening time (normally 9:30:00). If no eligible trade is received by 5 minutes after the opening time, no opening price will be reported. The opening price will be delayed by 15 minutes.
- For non-NEO listed securities, the closing price of a security is the last round lot trade received from the security's listing exchange when any timestamp exceeds the end of the listing exchange's regular session (normally 16:00:00). If no eligible trade has occurred prior to the close, no closing price will be reported from a Cboe market. The closing price will be delayed by 15 minutes.

The ***Market Center*** will identify the Cboe exchange that set the Opening or Closing Price for this security.

Table 22. Opening/Closing Price Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*m*	Alphanumeric	OP

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
timestamp	*ts*	Integer	The time the eligible trade occurred on the specified Market Center. Encoded as the number of nanoseconds since midnight.
symbol	*sy*	Alphanumeric	Relevant symbol.
marketCenter	*mc*	Alphanumeric	Y = BYX (US only) Z = BZX (US only) A = EDGA (US only) X = EDGX (US only) C = CTA (US only) U = UTP (US only) M = MATCHNow(Canada only) L = NEO-L (Canada only) N = NEO-N(Canada only) D = NEO-D(Canada only) T = TSX (Canada only) V = TSXV (Canada only) S = CSE (Canada only) r = NEO-Cross (Canada only) t = NEO-SST (Canada only)
openCloseIndicator	*OC*	Alphanumeric	O = *Price* is the Opening price C = *Price* is the Closing price
Price	*p*	Price	Opening/Closing Price.

End of Day Summary Message Fields

The **`End of Day Summary`** message is used to give a consolidated, high-level overview of a security for the day, based on an end of day summary message received from either the US CTA or UTP Securities Information Processor (SIP) feeds. The Canadian summary will be calculated using data from exchanges in the TMX IP feed. The message will be sent after it is received from the source and the delay period has expired, currently 15 minutes for US. Canada's summary will be sent at 17:15:00 for all symbols. The system may send more than one **`End of Day Summary`** message in the US after the end of a trading session due to multiple CTA/UTP end of day summary spins.

Table 23. End of Day Summary Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*m*	Alphanumeric	DS
timestamp	*ts*	Integer	The time the eligible trade occurred on the specified Market Center. Encoded as the number of nanoseconds since midnight.
symbol	*sy*	Alphanumeric	Relevant symbol.
marketCenter	*mc*	Alphanumeric	C = CTA US only U = UTP US only i = Indicative value Canada only c = Calculated value (from TMX IP) Canada only
openPrice	*oP*	Price	Opening Price. "0" if the system has not received an opening price for the security.
closePrice	*cP*	Price	Closing Price.
highPrice	*hP*	Price	Highest trade price of the day.
lowPrice	*lP*	Price	Lowest trade price of the day.
nationalVolume	*sv*	Integer	In the US, cumulative number of shares traded today as reported to the CTA and UTP SIPs. In Canada, the sum of two values: the current Cboe Cumulative Executed Volume and the 15-minute delayed executed volume of other Canadian exchanges as reported by TMX IP.

Cboe One Feed Example Messages

Clear Quote Message Example


```
{
 "m": "CQ",
 "ts": 1,
 "sy": "AAPL",
 "mc": "*",
 "s": 3
}
```

Symbol Summary Message Example


```
{
 "m": "S",
 "ts": 1,
 "sy": "AAPL",
 "cv": 100,
 "Bb": "1.00",
 "Bs": 100,
 "Ba": "0.90",
 "As": 100,
 "sv": 100,
 "f": 0,
 "s": 52
}
```

Best Quote Update Message Example


```
{
 "m": "Q",
 "ts": 1,
 "sy": "AAPL",
 "sd": "B",
 "P": "1.00",
 "S": 100,
 "s": 5
}
```

Cboe Market Status Message Example


```
{
 "m": "MS",
 "ts": 1,
 "mc": "Z",
 "ms": "N",
 "s": 22
}
```

ADAP Message Example


```
{
 "m": "A",
 "ts": 1,
 "sy": "AAPL",
 "f": 0,
 "ab": [
 {
  "mc": "Y",
  "sd": "B",
  "p": "1.00",
  "sz": 100
 },
 {
  "mc": "Y",
  "sd": "S",
  "p": "0.90",
  "sz": 100
 }
 ],
 "s": 109
}
```

Retail Price Improvement (RPI) Message Example


```
{
 "m": "RP",
 "ts": 1,
 "sy": "AAPL",
 "mc": "A",
 "rn": "B",
 "s": 144
}
```

Trade Message Example


```
{
 "m": "T",
 "ts": 1,
 "sy": "AAPL",
 "mc": "X",
 "e": "0000EXEC1",
 "lp": "1.00",
 "ls": 100,
 "cv": 200,
 "sv": 300,
 "f": 0,
 "s": 157
}
```

Trade Break Message Example


```
{
 "m": "TB",
 "ts": 1,
 "sy": "AAPL",
 "mc": "A",
 "e": "0000EXEC1",
 "cv": 200,
 "sv": 300,
 "f": 0,
 "s": 191
}
```

Trading Status Message Example


```
{
 "m": "TS",
 "ts": 1,
 "sy": "AAPL",
 "mc": "Y",
 "h": "H",
 "rs": "0",
 "s": 267
}
```

Opening/Closing Price Message Example


```
{
 "m": "OP",
 "ts": 1,
 "sy": "AAPL",
 "mc": "Z",
 "OC": "O",
 "p": "1.00",
 "s": 300
}
```

End of Day Summary Message Example


```
{
 "m": "DS",
 "ts": 1,
 "sy": "AAPL",
 "mc": "U",
 "oP": "1.00",
 "cP": "3.00",
 "hP": "4.00",
 "lP": "0.50",
 "sv": 500,
 "s": 1352
}
```

Cboe Equities TOP Feed (BZX, EDGX, BXE, CXE, DXE)

Overview

Cboe members may use the Cboe Global Cloud Equities TOP (TOP) feed to receive real-time top of book quotations and execution information from the following Equities Exchanges:

- BZX and EDGX US Equities Exchange platforms
- BXE, CXE, and DXE EU Equities Exchange platforms

The quotations received via TOP provide an aggregated size and do not indicate the size or number of individual orders at the best bid or ask. The TOP protocol also provides last trade price and size and cumulative volume data.

TOP cannot be used to enter orders. For order entry, refer to the appropriate specification below.

- Cboe US Equities FIX and BOE
- Cboe EU Equities FIX and BOE

Symbol Ranges and Units

Symbols will be separated into units by a published alphabetical distribution. Symbol distribution will not change intra-day. Cboe does, however, reserve the right to change the symbol distribution with prior notice to members. Care should be taken to ensure that symbol distribution changes can be supported easily.

The Kafka topic for the TOP feed is partitioned by unit, which means that messages from the same unit will be received in the order they were produced.

Message Format

The TOP feed is comprised of a series of Kafka JSON messages. Each message has a messageType (m) property that indicates which fields are on the message. Fields are properties that conform to the standard JSON pattern field name: field value. Field values must be one of the data types in Data Types on page 52.

Data Types

The TOP feed provides the same messages as the Cboe US Equities TOP feed but in a JSON format. Within the JSON format the following format is used for each data type.

- **Numeric** fields are a string of ASCII digits that can be converted to an integer.
- **Alpha** and **Alphanumeric** represent one or more printable ASCII characters.
- **Price** is a string of ASCII digits that can be converted to a floating point number. For example, the floating point number 12.34 is represented as `12.34`.

TOP Messages

Bid Update Message Fields

`Bid Update` messages replace the previous Top of Book bid for a symbol. One **`Bid Update`** message may reflect one or more updates to the inside bid that were processed at the same time.

Table 24. Bid Update Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
messageType	*m*	"E"	**`Bid Update`** message
timestamp	*t*	Numeric	Milliseconds since midnight ET (BZX, EDGX), or Milliseconds since midnight London (BXE, CXE, DXE)
symbol	*s*	Alpha	Symbol
bidPrice	*bp*	Price	Bid price; 0 if no bids
bidSize	*bs*	Numeric	Bid size; 0 if no bids

Ask Update Message Fields

`Ask Update` messages replace the previous Top of Book ask for a symbol. One **`Ask Update`** message may reflect one or more updates to the inside ask that were processed at the same time.

Table 25. Ask Update Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
messageType	*m*	"e"	**`Ask Update`** message
timestamp	*t*	Numeric	Milliseconds since midnight ET (BZX, EDGX), or Milliseconds since midnight London (BXE, CXE, DXE)
symbol	*s*	Alpha	Symbol
askPrice	*ap*	Price	Ask price; 0 if no asks
askSize	*as*	Numeric	Ask size; 0 if no asks

Two-Sided Update Message Fields

`Two-Sided Update` messages replace the previous Top of Book bid and ask for a symbol. One **`Two-Sided Update`** message may reflect one or more updates to the inside bid/ask that were processed at the same time.

Table 26. Two-Sided Update Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
messageType	*M*	"F"	**`Two-Sided Update`** message
timestamp	*T*	Numeric	Milliseconds since midnight ET (BZX, EDGX), or Milliseconds since midnight London (BXE, CXE, DXE)
symbol	*S*	Alpha	Symbol
bidPrice	*Bp*	Price	Bid price; 0 if no bids
bidSize	*Bs*	Numeric	Bid size; 0 if no bids
askPrice	*ap*	Price	Ask price; 0 if no asks
askSize	*as*	Numeric	Ask size; 0 if no asks

Trade Message Fields

When an execution occurs, a **`Trade`** message is sent which includes the cumulative volume and last price and size. Multiple executions which occur concurrently will be compressed into a single update for bandwidth savings. TOP does not send a trade message for every individual fill.

In the event of a trade break or correction, it is possible that the cumulative volume will decrease from the previous update.

Please note that for EU markets (BXE, CXE, DXE) the *lastTradePrice*, *lastTradeSize*, and *volume* fields relate only to lit-book executions on the applicable venue. Cboe provides a separate Last Sale feed containing the full trade type granularity.

Table 27. Trade Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
messageType	*m*	"f"	**`Trade`** message
timestamp	*t*	Numeric	Milliseconds since midnight ET (BZX, EDGX), or Milliseconds since midnight London (BXE, CXE, DXE)
symbol	*s*	Alpha	Symbol
lastTradePrice	*tp*	Price	Last trade price
lastTradeSize	*ts*	Numeric	Last trade size
volume	*v*	Numeric	Cumulative number of shares traded today

Trading Status Message Fields

The **`Trading Status`** message is used to indicate the current trading status of a security. A **`Trading Status`** message will be sent whenever a security's trading status changes.

A *haltStatus* of S will be implied at system startup. T will be sent as securities are available for trading. A will be distributed when orders can be accepted for queuing in preparation for the market open.

Table 28. Trading Status Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
messageType	*m*	't'	**`Trading Status`** Message
timestamp	*t*	Numeric	Milliseconds since midnight ET (BZX, EDGX), or Milliseconds since midnight London (BXE, CXE, DXE)
symbol	*s*	Alpha	Symbol
haltStatus	*h*	Alpha	A = Accepting Orders for Queuing H = Halted Q = Quote-Only (Cboe Listings) S = Exchange Specific Suspension T = Trading
regShoAction	*r*	Alphanumeric	0 = No price test in effect 1 = Reg SHO price test restriction in effect

Trading Status Message Fields for BXE, CXE and DXE (EU)

The **`Trading Status`** message is used to indicate the current trading status of a security. A **`Trading Status`** message will be sent whenever a security's trading status changes.

Table 29. Trading Status Message Fields for BXE, CXE and DXE (EU)

FIELD NAME	JSON	VALUE	DESCRIPTION
messageType	*m*	't'	**`Trading Status`** message
timestamp	*t*	Numeric	Milliseconds since midnight ET (BZX, EDGX), or Milliseconds since midnight London (BXE, CXE, DXE)
symbol	*s*	Alpha	Symbol
haltStatus	*h*	Alpha	T = Trading R = Off-Book Reporting C = Closed S = Suspension N = No Reference Price V = Volatility Interruption O = Opening Auction E = Closing Auction M = Market Order Imbalance Extension P = Price Monitoring Extension U = Cboe Closing Cross

TOP Example Messages

The following are examples of each message type that can be sent on TOP.

Bid Update Message Example

```
{
 "m":"E",
 "t": 36000123
 "s": "ABCDE",
 "bp": "320.19",
 "bs": 1000
}
```

Ask Update Message Example

```
{
 "m": "e",
 "t": 36000123
 "s": "ABCDE",
 "ap": "320.21",
 "as": 200
}
```

Two-Sided Update Message Example


```
{
 "m": "F",
 "t": 36000123,
 "s": "ABCDE",
 "bp": "320.19",
 "bs": 1000,
 "ap": "320.21",
 "as": 200
}
```

Trade Message Example


```
{
 "m": "f",
 "t": 36000123,
 "s": "ABCDE",
 "tp": "320.20",
 "ts": 150,
 "v": 150
}
```

Trading Status Message Example for BZX and EDGX (US)

```
{
 "m": "t",
 "t": 36000123,
 "s": "ABCDE",
 "h": "H",
 "r": "0"
}
```

Trading Status Message Example for BXE, CXE and DXE (EU)

```
{
 "m": "t",
 "t": 36000123,
 "s": "VODl",
 "h": "T"
}
```

Cboe AU Equities TOP Feed

Overview

Cboe customers may use the Cboe Global Cloud AU Equities TOP (TOP) feed to receive real-time top of book quotations and execution information from the CXA Equities Exchange platform.

The quotations received via TOP provide an aggregated size and do not indicate the size or number of individual orders at the best bid or ask. The TOP protocol also provides last trade price and size and cumulative volume data.

TOP cannot be used to enter orders. For order entry, refer to the appropriate CXA Equities FIX or BOE specification.

Symbol Ranges and Units

Symbols will be separated into units by a published alphabetical distribution. Symbol distribution will not change intra-day. Cboe does, however, reserve the right to change the symbol distribution with prior notice to members. Care should be taken to ensure that symbol distribution changes can be supported easily.

The Kafka topic for the TOP feed is partitioned by unit, which means that messages from the same unit will be received in the order they were produced.

Message Format

The TOP feed is comprised of a series of Kafka JSON messages. Each message has a messageType (m) property that indicates which fields are on the message. Fields are properties that conform to the standard JSON pattern field name: field value. Field values must be one of the data types in Data Types on page 68.

Data Types

The TOP feed provides the same messages as the Cboe US Equities TOP feed but in a JSON format. Within the JSON format the following format is used for each data type.

- **Numeric** fields are a string of ASCII digits that can be converted to an integer.
- **Alpha** and **Alphanumeric** represent one or more printable ASCII characters.
- **Price** is a string of ASCII digits that can be converted to a floating point number. For example, the floating point number 12.34 is represented as "12.34".
- **Bit Field** fields are fixed width fields with each bit representing a Boolean flag (the 0 bit is the lowest significant bit; the 7 bit is the highest significant bit).

Sequence Field

The *Sequence* field is used for all TOP messages delivered via Amazon MSK. Examples can be found in the messages below. Messages are sequenced on a per-unit basis.

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Sequence	*s*	Integer	Sequence representing the order that the messages were received by the Kafka producer.

TOP Messages

Unit Clear Message Fields

The **`Unit Clear`** message instructs feed recipients to clear all market snapshots for the TOP book in the partition. It will be distributed in rare recovery events such as a data center fail-over.

Table 30. Unit Clear Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	"U"	Unit Clear message.
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)

Trading Status Message Fields

The **`Trading Status`** message is used to indicate the current trading status of a security. A **`Trading Status`** message will be sent whenever trading status changes for a security. The following summarizes the *Trading Status* values in the CXA system:

- H = Halt state. Not accepting orders or off-exchange trade reports, though existing orders may be canceled. Implied at system startup for all symbols.
- A = Pre-market. Not accepting orders, but off-exchange trades may be reported.
- T=Trading. Continuous trading session open. Accepting orders and off-exchange trade reports.
- M=MOC Trading. Continuous trading session closed. Accepting only MOC orders and off-exchange trade reports.
- P = Post-market. MOC only trading session closed. Not accepting orders, but off-exchange trades may be reported.
- C = Closed. Not accepting orders or off-exchange trade reports.
- S = Trading suspended. Sent in the event that trading is suspended for operational reasons. Not accepting orders or off-exchange trade reports, though existing orders may be canceled.
- O = Pre-Open. Symbols eligible for the CXA Opening Auction, or are subject to an intra-day Auction, are placed into Pre-Open prior to the auction. Order entry, amends, cancels are permitted. Orders are not matched and overlapping order books may form.
- E = Pre-Close. Symbols eligible for the CXA Closing Auction are placed into Pre-Close prior to the auction. Order entry, amends, cancels are permitted. Orders are not matched and overlapping order books may form.

Halt and Trading suspended are functionally the same, though a halt is considered short term while suspended occurs for a longer term that can persist over several days.

The *Trading Status* field will be used to represent the status of the trading session.

Table 31. Trading Status Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*M*	"t"	**`Trading Status`** Message
Timestamp	*Tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*Sy*	Alphanumeric	Symbol
Trading Status	*H*	Alpha	H = Halted A = Pre-market T = Trading M = MOC Trading P = Post-market C = Closed S = Trading suspended O = Pre-Open E = Pre-Close

FIELD NAME	JSON	VALUE	DESCRIPTION
Market ID Code	*mC*	Alpha	Market Identifier Codes: XASX = Australian Stock Market CXAW = CXA Warrants CXAE = CXA ETF CXAQ = CXA QMF CXAL = CXA Listed

Bid Update Message Fields

`Bid Update` messages replace the previous Top of Book bid for a symbol. One **`Bid Update`** message may reflect one or more updates to the inside bid that were processed at the same time.

Table 32. Bid Update Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"E"	**`Bid Update`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Bid Price	*bP*	Price	Bid price; 0 if no bids
Bid Quantity	*bS*	Numeric	Bid size; 0 if no bids or only undisclosed volume is at this price

Ask Update Message Fields

`Ask Update` messages replace the previous Top of Book ask for a symbol. One **`Ask Update`** message may reflect one or more updates to the inside ask that were processed at the same time.

Table 33. Ask Update Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"e"	**`Ask Update`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Ask Price	*aP*	Price	Ask price; 0 if no asks
Ask Quantity	*aS*	Numeric	Ask size; 0 if no asks or only undisclosed volume is at this price

Two-Sided Update Message Fields

`Two-Sided Update` messages replace the previous Top of Book bid and ask for a symbol. One **`Two-Sided Update`** message may reflect one or more updates to the inside bid/ask that were processed at the same time.

Table 34. Two-Sided Update Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"F"	**`Two-Sided Update`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Bid Price	*bP*	Price	Bid price; 0 if no bids
Bid Quantity	*bS*	Numeric	Bid size; 0 if no bids or only undisclosed volume is at this price
Ask Price	*aP*	Price	Ask price; 0 if no asks
Ask Quantity	*aS*	Numeric	Ask size; 0 if no asks or only undisclosed volume is at this price

Trade Message Fields

When an execution occurs electronically on the CXA book or off-exchange and reported to CXA, then a **`Trade`** message is sent which includes the cumulative volume and last price and size. **`Trade`** messages do not alter the book. For on-exchange electronic executions one or more **`Bid/Ask Update`** or **`Two-Sided Update`** messages may follow a **`Trade`** message to reflect the updated book (for example, an aggressive order may take out one or more price levels and establish a new level on the opposite side).

Any order may be executed in parts. A complete view of all CXA executions can be built from all **`Trade`** messages for on-exchange electronic executions.

A **`Trade`** message is also sent whenever an execution or trade report is broken with the *Trade Condition* field value indicating trade break. Trade breaks are rare and only affect applications that rely upon CXA execution-based data. Trade breaks will contain the *Symbol*, *Quantity*, *Price*, and *Execution ID* of the original trade. The *Volume* field will be reduced by the number of shares reported in the *Trade Size* field.

Table 35. Trade Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*M*	"f"	**`Trade`** message
Timestamp	*Tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*Sy*	Alphanumeric	Symbol
Trade Price	*tp*	Price	Last trade price
Trade Quantity	*sz*	Numeric	Last trade size
Execution ID	*e*	Alphanumeric	CXA assigned day-unique execution identifier of this trade or trade break.
Total Volume	*v*	Numeric	Cumulative number of shares traded today
PID	*PA*	Numeric	4 digit Participant ID, or empty ("") if not attributed
Contra PID	*CO*	Numeric	4 digit Contra Participant ID, or empty ("") if not attributed
Trade Type	*Tt*	Alpha	B = Broker Preferenced Trade N = Trade resulting from normal matching logic O = Opening Auction Trade or Intra-day Auction Trade C = Closing Auction Trade H = Halt Auction (Re-Opening) <space> = Off-exchange trade report
Trade Designation	*Td*	Alpha	C = CXAC (Limit) P = CXAP (Mid Point) N = CXAN (Near Point) F = CXAF (Far Point) M = CXAM (MOC) Valid only for on-exchange executions, space otherwise.
Trade Report Type	*rt*	Alpha	B = Block Trade P = Large Portfolio Trade

FIELD NAME	JSON	VALUE	DESCRIPTION
			T = Large Principal Transaction S = Trades With Price Improvement L = Permitted Trade During Post Trading Hours Period M = Permitted Trade During Pre Trading Hours Period E = Out Of Hours Trade F = ETF Trade Report for unit creations or redemptions
Trade Condition	*tc*	Alpha	N = Normal X = Trade Break
flags	*f*	Bit Field	Bit0 (Value 1) - Reserved Bit1 (Value 2) - Trade from Converted Order Bit2 (Value 4) - Reserved Bit3 (Value 8) - Reserved Bit4 (Value 16) - Reserved Bit5 (Value 32) - Reserved Bit6 (Value 64) - Reserved Bit7 (Value 128) - Reserved

Calculated Value Message Fields

The **`Calculated Value`** message is sent when CXA calculates market values for a specified symbol or when a calculated market value is reported to CXA. The specified symbol may not trade on CXA, but instead could represent index or iNAV values reported to CXA from third parties as indicated by the ***Value Category*** field. The index values will be reported on each of the unitized CXA TOP feeds and are not specific to an individual unit.

Table 36. Calculated Value Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"F"	**`Calculated Value`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Value category	*Vc*	Numeric	`1` = Closing price `2` = iNAV values (ETF) `3` = Index values `4` = EOD NAV from issuer
Value	*Cv*	Price	The calculated value
Value Timestamp	*vt*	Numeric	Timestamp when the calculated value was generated (microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC).

End of Session Message Fields

The **`End of Session`** message is sent for each partition when the matching unit for that partition shuts down. No additional sequenced messages will be delivered on this partition.

Table 37. End of Session Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"S"	**`End of session`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)

Auction Update Message Fields

The **`Auction Update`** message is sent during Pre-Open or Pre-Close trading states when a symbol has an order update which relates to an overlapping order book. The message includes the potential number of shares and the indicative auction price if the symbol were to open now.

Table 38. Auction Update Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*		**`Auction Update`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Auction Type	*a*	Alpha	O = Opening / Intra-day Auction C = Closing Auction H = Halt Auction (Re-Opening)
Buy Shares	*bs*	Numeric	Number of shares on buy side eligible to trade at the auction indicative price.
Sell Shares	*ss*	Numeric	Number of shares on sell side eligible to trade at the auction indicative price.
Indicative Price	*ip*	Price	Indicative Auction Price at which the continuous book would match.

Auction Summary Message Fields

The **`Auction Summary`** message is sent at the completion of the Auction and includes the Auction Price and Shares traded, per symbol.

Table 39. Auction Summary Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*		**`Auction Summary`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Auction Type	*a*	Alpha	O = Opening / Intra-day Auction C = Closing Auction H = Halt Auction (Re-Opening)
Price	*p*	Price	Auction price
Shares	*sh*	Numeric	Cumulative number of shares executed during the auction.

Cboe AU Equities PITCH Feed

Overview

Cboe members may use the Cboe Global Cloud AU Equities PITCH (PITCH) feed to receive real-time order-by-order and execution information from the CXA Equities Exchange platform.

PITCH cannot be used to enter orders. For order entry, refer to the appropriate CXA Equities FIX or BOE specification.

Symbol Ranges and Units

Symbols will be separated into units by a published distribution. Symbol distribution will not change intra-day. Cboe does, however, reserve the right to change the symbol distribution with prior notice to clients. Care should be taken to ensure that symbol distribution changes can be supported easily.

The Kafka topic for the PITCH feed is partitioned by unit, which means that messages from the same unit will be received in the order they were produced.

Message Format

The PITCH feed is comprised of a series of Kafka JSON messages. Each message has a messageType (m) property that indicates which fields are on the message. Fields are properties that conform to the standard JSON pattern field name: field value. Field values must be one of the data types in Data Types on page 85.

Data Types

The PITCH feed provides the same messages as the Cboe US Equities PITCH feed but in a JSON format. Within the JSON format the following format is used for each data type.

- **Numeric** fields are a string of ASCII digits that can be converted to an integer.
- **Alpha** and **Alphanumeric** represent one or more printable ASCII characters.
- **Price** is a string of ASCII digits that can be converted to a floating point number. For example, the floating point number 12.34 is represented as "12.34".
- **Bit Field** fields are fixed width fields with each bit representing a Boolean flag (the 0 bit is the lowest significant bit; the 7 bit is the highest significant bit).

Sequence Field

The *Sequence* field is used for all PITCH messages delivered via Amazon MSK. Examples can be found in the messages below. Messages are sequenced on a per-unit basis.

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Sequence	*s*	Integer	Sequence representing the order that the messages were received by the Kafka producer.

PITCH Messages

Unit Clear Message Fields

The **`Unit Clear`** message instructs feed recipients to clear all market snapshots for the PITCH book in the partition. It will be distributed in rare recovery events such as a data center fail-over.

Table 40. Unit Clear Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	"U"	**`Unit Clear`** message.
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)

Trading Status Message Fields

The **`Trading Status`** message is used to indicate the current trading status of a security. A **`Trading Status`** message will be sent whenever trading status changes for a security. The following summarises the *Trading Status* values in the CXA system:

- C = Closed. Not accepting orders or off-exchange trade reports. Implied at system start-up for all symbols.
- A = Pre-market. Not accepting orders, off-exchange trades may be reported.
- T = Trading. Continuous trading session open. Accepting orders and off-exchange trade reports.
- M = MOC Trading. Continuous trading session closed. Accepting only MOC orders and off-exchange trade reports.
- P = Post-market. MOC only trading session closed. Not accepting orders, off-exchange trades may be reported.
- H = Halted. Not accepting orders, only eligible off-exchange trades may be reported. Existing orders may be cancelled.
- S = Trading suspended. Sent in the event trading is suspended for operational reasons. Not accepting orders, only eligible off-exchange trades may be reported. Existing orders may be cancelled.
- O = Pre-Open. Symbols eligible for the CXA Opening Auction, or are subject to an intra-day Auction, are placed into Pre-Open prior to the auction. Order entry, amends, cancels are permitted. Orders are not matched and overlapping order books may form.
- E = Pre-Close. Symbols eligible for the CXA Closing Auction are placed into Pre-Close prior to the auction. Order entry, amends, cancels are permitted. Orders are not matched and overlapping order books may form.

Halted and Trading suspended are functionally the same, though a halt is considered short term while a suspension occurs for a longer term that can persist over several days.

The *Trading Status* field will be used to represent the status of the trading session.

Table 41. Trading Status Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"t"	**`Trading Status`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Trading Status	*h*	Alpha	H = Halted A = Pre-market T = Trading M = MOC Trading P = Post-market

FIELD NAME	JSON	VALUE	DESCRIPTION
			C = Closed S = Trading suspended O = Pre-Open E = Pre-Close
Market ID Code	*mC*	Alpha	Market Identifier Codes: XSAX = Australian Stock Market CXAW = CXA Warrants CXAE = CXA ETF CXAQ = CXA QMF CXAL = CXA Listed

Add Order Message Fields

The **`Add Order`** message represents a newly accepted visible or undisclosed order on the CXA book. It includes a day-specific ***Order Id*** assigned by CXA to the order.

Table 42. Add Order Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"A"	**`Add Order`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Price	*p*	Price	Displayed price for the order
Quantity	*sz*	Numeric	Displayed size of the order; 0 if undisclosed.
Side	*sd*	Alpha	`B` = buy `S` = sell
PID	*PA*	Alphanumeric	4 digit Participant ID, or empty ("") if not attributed
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"

Order Executed Message Fields

`The Order Executed` message is sent when a visible order on the CXA book is executed in whole or in part. The execution price equals the order price found in the original **`Add Order`** message or the order price in the latest **`Modify Order`** message referencing the *Order Id*.

Table 43. Order Executed Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"X"	**`Order Executed`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Price	*p*	Price	Displayed price for the order
Execution ID	*e*	Alphanumeric	CXA assigned day-unique execution identifier of this trade or trade break.
Executed Quantity	*es*	Numeric	Executed Shares
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"
Contra PID	*CO*	Alphanumeric	4-digit Contra Participant ID, or empty ("") if not attributed
Contra Order ID	*Co*	Alphanumeric	Day-specific identifier assigned to the contra order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOX"
Trade Type	*Tt*	Alpha	N = Continuous O = Opening / Intra-day Auction C = Closing Auction H = Halt Auction (Re-Opening)

Reduce Size Message Fields

`The Reduce Size` message is sent when a visible order on the CXA book is partially reduced.

Table 44. Reduce Size Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"R"	**`Reduce Size`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"
Cancelled Quantity	*cs*	Numeric	Size removed from the order.

Modify Order Message Fields

The **`Modify Order`** message is sent whenever an open order is visibly modified. The *Order Id* refers to the *Order Id* of the original **`Add Order`** message.

Note that **`Modify Order`** messages that appear to be no ops (i.e., they do not appear to modify any relevant fields) will still lose priority.

Table 45. Modify Order Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"M"	**`Modify Order`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"
Price	*p*	Price	Displayed price for the order
Quantity	*sz*	Numeric	Displayed size of the order; 0 if undisclosed.

Delete Order Message Fields

The **`Delete Order`** message is sent whenever a booked order is cancelled or leaves the order book. The *Order Id* refers to the *Order Id* of the original **`Add Order`** message.

Although not currently possible, in the future under certain circumstances an order that is deleted from the book may return to the book later. Therefore, a **`Delete Order`** message does not indicate that a given *Order Id* will not be sent again on a subsequent **`Add Order`** message. Participants should be prepared to handle this scenario.

Table 46. Delete Order Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"D"	**`Delete Order`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"

Trade Message Fields

The **`Trade`** message provides information about executions of non-displayed and undisclosed orders on the CXA book or executions that occur off-exchange and reported to CXA. **`Trade`** messages for on-exchange electronic executions are necessary to calculate CXA execution-based data. **`Trade`** messages do not alter the book and can be ignored if messages are being used solely to build a book.

No **`Add Order`** message is sent for hidden orders, and thus, no modify order messages may be sent when hidden orders are executed. Instead, a **`Trade`** message for on-exchange electronic executions is sent whenever a hidden order is executed in whole or in part. A **`Trade`** message for on-exchange electronic executions is also sent when there is an execution against any non-displayed portion of an iceberg order. As with visible orders, hidden and iceberg orders may be executed in parts. **A complete view of all CXA executions can be built by combining allOrder Executed and Trade messages.**

Table 47. Trade Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"f"	**`Trade`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Trade Price	*tp*	Price	trade price
Trade Quantity	*sz*	Numeric	trade size
Execution ID	*e*	Alphanumeric	CXA assigned day-unique execution identifier of this trade or trade break.
Trade Type	*Tt*	Alpha	<space> = Off-exchange trade B = Broker Preferenced Trade N = Trade resulting from normal matching logic O = Opening Auction Trade/Intra-day Auction Trade C = Closing Auction Trade H = Halt Auction (Reopening)
Trade Designation	*Td*	Alpha	Valid only for on-exchange executions, space otherwise. <space> = Off-exchange trade C = CXAC (Limit) P = CXAP (Mid Point) N = CXAN (Near Point) F = CXAF (Far Point) M = CXAM (MOC) Valid only for on-exchange executions, space otherwise.
Trade Report Type	*rt*	Alpha	Valid only for off-exchange trade reports, space otherwise. <space> = Regular trade B = Block Trade

FIELD NAME	JSON	VALUE	DESCRIPTION
			P = Large Portfolio Trade T = Large Principal Transaction S = Trades With Price Improvement L = Permitted Trade During Post Trading Hours Period M = Permitted Trade During Pre Trading Hours Period E = Out Of Hours Trade F = ETF Trade Report for unit creations or redemptions
Trade Transaction Time	*tt*	Numeric	Nanosecond timestamp since epoch (1/1/70 00:00 UTC). Zero if *Trade Type* is B or N.
PID	*PA*	Numeric	4 digit Participant ID, or empty ("") if not attributed
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"
Contra PID	*CO*	Numeric	4 digit Contra Participant ID, or empty ("") if not attributed
Contra Order ID	*Co*	Alphanumeric	Day-specific identifier assigned to the contra order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOX"
flags	*f*	Bit Field	Bit0 (Value 1) - Reserved Bit1 (Value 2) - Trade from Converted Order Bit2 (Value 4) - Reserved Bit3 (Value 8) - Reserved Bit4 (Value 16) - Reserved Bit5 (Value 32) - Reserved Bit6 (Value 64) - Reserved Bit7 (Value 128) - Reserved

Trade Break Message Fields

The **`Trade Break`** message is sent whenever an execution on CXA or off-exchange trade reported to CXA is cancelled. A trade correction is performed by sending a **`Trade Break`** message followed by a new **`Trade`** message with the corrected size and price. Applications that simply build a CXA book can ignore **`Trade Break`** messages.

Table 48. Trade Break Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"x"	**`Trade Break`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Execution ID	*e*	Alphanumeric	CXA assigned day-unique execution identifier of this trade or trade break.

Calculated Value Message Fields

The **`Calculated Value`** message is sent when CXA calculates market values for a specified symbol or when a calculated market value is reported to CXA. The specified symbol may not trade on CXA, but instead could represent index values or NAV reported to CXA from third parties as indicated by the ***Value Category*** field.

Table 49. Calculated Value Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"F"	**`Calculated Value`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Value category	*Vc*	Numeric	`1` = Closing price `2` = iNAV values (ETF) `3` = Index values `4` = EOD NAV from issuer
Value	*Cv*	Price	The calculated value
Value Timestamp	*vt*	Numeric	Timestamp when the calculated value was generated (microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC).

End of Session Message Fields

The **`End of Session`** message is sent for each partition when the matching unit for that partition shuts down. No additional sequenced messages will be delivered on this partition.

Table 50. End of Session Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"S"	**`End of Session`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)

Auction Update Message Fields

The **`Auction Update`** message is sent during Pre-Open or Pre-Close trading states when a symbol has an order update which relates to an overlapping order book. The message includes the potential number of shares and the indicative auction price if the symbol were to open now.

Table 51. Auction Update Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	AU	**`Auction Update`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Auction Type	*a*	Alpha	O = Opening / Intra-day Auction C = Closing Auction H = Halt Auction (Re-Opening)
Buy Shares	*bs*	Numeric	Number of shares on buy side eligible to trade at the auction indicative price.
Sell Shares	*ss*	Numeric	Number of shares on sell side eligible to trade at the auction indicative price.
Indicative Price	*ip*	Price	Indicative Auction Price at which the continuous book would match.

Auction Summary Message Fields

The **`Auction Summary`** message is sent at the completion of the Auction and includes the Auction *Price* and *Shares* traded, per symbol.

Table 52. Auction Summary Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	AS	**`Auction Summary`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol (right padded with spaces).
Auction Type	*a*	Alpha	O = Opening / Intra-day Auction C = Closing Auction H = Halt Auction (Re-Opening)
Price	*p*	Price	Auction price
Shares	*sh*	Numeric	Cumulative number of shares executed during the auction.

Cboe JP Equities PITCH Feed

Overview

Cboe members may use the Cboe Global Cloud JP Equities PITCH (PITCH) feed to receive real-time order-by-order and execution information from the CXJ Alpha (CXJA) and CXJ Select (CXJS) Equities Exchange platform.

PITCH cannot be used to enter orders. For order entry, refer to the appropriate CXJ Equities FIX or BOE specification.

Symbol Ranges and Units

Symbols will be separated into units by a published distribution. Symbol distribution will not change intra-day. Cboe does, however, reserve the right to change the symbol distribution with prior notice to clients. Care should be taken to ensure that symbol distribution changes can be supported easily.

The Kafka topic for the PITCH feed is partitioned by unit, meaning messages from the same unit will be received in the order they were produced.

Message Format

The PITCH feed is comprised of a series of Kafka JSON messages. Each message has a messageType (m) property indicating which fields are on the message. Fields are properties that conform to the standard JSON pattern field name: field value. Field values must be one of the data types in Data Types on page 85.

Data Types

The PITCH feed provides the same messages as the Cboe US Equities PITCH feed but in a JSON format. Within the JSON format the following format is used for each data type.

- **Numeric** fields are a string of ASCII digits that can be converted to an integer.
- **Alpha** and **Alphanumeric** represent one or more printable ASCII characters.
- **Price** is a string of ASCII digits that can be converted to a floating point number. For example, the floating point number 12.34 is represented as `12.34`.

Sequence Field

The *Sequence* field is used for all PITCH messages delivered via Amazon MSK. Examples can be found in the messages below. Messages are sequenced on a per-unit basis.

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Sequence	*s*	Integer	Sequence representing the order that the messages were received by the Kafka producer.

PITCH Messages

Unit Clear Message Fields

The **`Unit Clear`** message instructs feed recipients to clear all market snapshots for the PITCH book in the partition. It will be distributed in rare recovery events such as a data center fail-over.

Table 53. Unit Clear Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	"U"	**`Unit Clear`** message.
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)

Trading Status Message Fields

The **`Trading Status`** message is used to indicate the current trading status of a security. A **`Trading Status`** message will be sent whenever trading status changes for a security. The following summarises the *Trading Status* values for CXJA and CXJS:

- C = Closed. Not accepting orders. Implied at system start-up for all symbols.
- h = Halted in Pre-market. (Halted).Not accepting orders, security is halted by the primary market before the scheduled start of trading. Note that a security which remains halted once the trading session starts will transition to H (Halted).
- T = Trading. Continuous trading session open. Accepting orders.
- H = Halted. Not accepting orders. Existing orders may be cancelled.

The *Trading Status* field is used to represent the status of the trading session.

Table 54. Trading Status Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"t"	**`Trading Status`** Message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Trading Status	*h*	Alpha	H = Halted h = Halted in Pre-market T = Trading C = Closed

Add Order Message Fields

The **`Add Order`** message represents a newly accepted visible or undisclosed order on the Cboe Japan book. It includes a day-specific *Order Id* assigned by Cboe Japan to the order.

Table 55. Add Order Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"A"	**`Add Order`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Price	*p*	Price	Displayed price for the order
Quantity	*sz*	Numeric	Displayed size of the order; 0 if undisclosed.
Side	*sd*	Alpha	`B` = buy `S` = sell
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"

Order Executed Message Fields

`The Order Executed` message is sent when a visible order on the Cboe Japan book is executed in whole or in part. The execution price equals the order price found in the original **`Add Order`** message or the order price in the latest **`Modify Order`** message referencing the Order Id.

Table 56. Order Executed Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"X"	**`Order Executed`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Execution ID	*e*	Alphanumeric	Day-unique execution identifier of this trade. Assigned by Cboe Japan.
Executed Quantity	*es*	Numeric	Executed Shares
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"
Contra Order ID	*Co*	Alphanumeric	Day-specific identifier assigned to the contra order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOX"
Tick Direction	*TD*	Alphanumeric	The tick direction of this trade. - = Lower than previous last sale + = Higher than previous last sale 0 = No change from or since TSE previous close (or first TSE trade of an IPO) D = Same as previous last sale, and most recent price change was downward ("-"). U = Same as previous last sale, and most recent price change was upward ("+").

Reduce Size Message Fields

`The Reduce Size` message is sent when a visible order on the Cboe Japan book is partially reduced.

Table 57. Reduce Size Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"R"	**`Reduce Size`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"
Cancelled Quantity	*cs*	Numeric	Size removed from the order.

Modify Order Message Fields

The **`Modify Order`** message is sent whenever an open order is visibly modified. The *Order Id* refers to the *Order Id* of the original **`Add Order`** message.

Note that **`Modify Order`** messages that appear to be no ops (i.e., they do not appear to modify any relevant fields) will still lose priority.

Table 58. Modify Order Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"M"	**`Modify Order`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"
Price	*p*	Price	Displayed price for the order
Quantity	*sz*	Numeric	Displayed size of the order; 0 if undisclosed.

Delete Order Message Fields

The **`Delete Order`** message is sent whenever a booked order is cancelled or leaves the order book. The *Order Id* refers to the *Order Id* of the original **`Add Order`** message.

Although not currently possible, in the future under certain circumstances an order that is deleted from the book may return to the book later. Therefore, a **`Delete Order`** message does not indicate that a given *Order Id* will not be sent again on a subsequent **`Add Order`** message. Participants should be prepared to handle this scenario.

Table 59. Delete Order Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"D"	**`Delete Order`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"

Trade Message Fields

The **`Trade`** message provides information about executions of non-displayed orders on the Cboe Japan book. **`Trade`** messages do not alter the book and can be ignored if messages are being used solely to build a book.

No **`Add Order`** message is sent for hidden orders, and thus, no modify order messages may be sent when hidden orders are executed. Instead, a **`Trade`** message for executions is sent whenever a hidden order is executed in whole or in part. A **`Trade`** message for executions is also sent when there is an execution against any non-displayed portion of an iceberg order. As with visible orders, hidden and iceberg orders may be executed in parts. A complete view of all executions can be built by combining all **`Order Executed`** and **`Trade`** messages.

Table 60. Trade Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"f"	**`Trade`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Symbol	*sy*	Alphanumeric	Symbol
Trade Price	*tp*	Price	trade price
Trade Quantity	*sz*	Numeric	trade size
Execution ID	*e*	Alphanumeric	Day-unique execution identifier of this trade. Assigned by Cboe Japan.
Order ID	*o*	Alphanumeric	Day-specific identifier assigned to this order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOV"
Contra Order ID	*Co*	Alphanumeric	Day-specific identifier assigned to the contra order, expressed in base36 (0-9, A-Z). Example: "874XH1UZEHOX"

Trade Break Message Fields

The **`Trade Break`** message is sent whenever an execution on Cboe Japan is cancelled. A trade correction is performed by sending a **`Trade Break`** message followed by a new **`Trade`** message with the corrected size and price. Applications that simply build a Cboe Japan book can ignore **`Trade Break`** messages.

Table 61. Trade Break Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"x"	**`Trade Break`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)
Execution ID	*e*	Alphanumeric	Day-unique execution identifier of the cancelled trade.

End of Session Message Fields

The **`End of Session`** message is sent for each partition when the matching unit for that partition shuts down. No additional sequenced messages will be delivered on this partition.

Table 62. End of Session Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
Message Type	*m*	"S"	**`End of Session`** message
Timestamp	*tu*	Numeric	Microseconds since epoch (Jan 1, 1970 00:00:00.00 UTC+0)

Cboe Futures TOP Feed

Overview

CFE participants may use the CFE TOP protocol to receive real-time top of book quotations direct from CFE.

The quotations received via TOP provide an aggregated size and do not indicate the size or number of individual orders at the best bid or ask. The TOP protocol also provides last trade price and size and cumulative volume data.

Feed Hours and System Restart

The TOP feed will start up on Sunday at approximately 10:00 a.m. CT and shutdown on Friday at approximately 4:05 p.m. CT. A daily restart occurs between 4:05 and 4:45 p.m. CT each day at which time sequences will be reset. The daily restart is typically observed between 4:05 and 4:10 p.m. CT but could occur later if needed for operational reasons. Feed startup and shutdown times may be adjusted without notice.

Under normal operations, it is expected that the order books will be cleared (including GTC and GTD orders), prior to the daily restart and reset of sequences. Persisted GTC and GTD orders will be added back onto the order books immediately after restart.

Symbol Ranges, Units, Sequence Numbers, and Timestamps

Products will be separated into units and messages pertaining to a specific unit will be sent to their own partition. Product distribution will not change intra-day. CFE does, however, **reserve the right change the product distribution with 48 hours prior notice to participants**. Care should be taken to ensure that product distribution changes can be supported easily.

Message sequence numbers are incremented by one for every sequenced message within a particular symbol unit. It is important to understand that one unit will be delivered per topic partition. As with symbol ranges, unit distribution across partitions will not change intra-day, but may change after notice has been given.

Messages in the TOP feed are timestamped relative to special **`Time`** messages sent once every second from each unit. Because ordering cannot be guaranteed between partitions, care must be taken to ensure that messages from a specific partition are compared ONLY to **`Time`** messages from that same partition.

Futures Specific Symbol Processing

CFE TOP feed **`Futures Instrument Definition`** and **`Futures Variance Symbol Mapping`** messages map simple futures contract and spread instruments to a six character, ASCII *Symbol*. For example, the weekly VX11 contract expiring March 14, 2017 might be represented by the *Symbol* `0abC12`. This symbol mapping significantly reduces the size of the Multicast TOP feed for futures and allows participants to use the same symbol handling mechanisms for the Cboe operated equity, options, and futures exchanges.

Mapping occurs on a continuous basis on each unit's partition. Unlike the Multicast TOP feed, **`Futures Instrument Definition`** and **`Futures Variance Symbol Mapping`** messages in Kafka TOP are all sequenced.

Spread instruments may be occasionally created intra-day. In these cases, the **`Futures Instrument Definition`** message will be sent as a sequenced message on the feed before any other messages that reference an instrument created intra-day are sent.

In addition to the symbol mapping messages available on the Multicast TOP feed, a downloadable file with current mappings is available via the CFE website at the CFE Contract Listings and Product Level System Parameters page.

Protocol

Message Format

The messages that make up the Futures TOP protocol are formatted with JSON, and each message contains the *Sequence* field (see Sequence Field on page 124), which handles sequencing and delivery integrity.

The Futures TOP data feed is comprised of a series of dynamic length sequenced messages. Each message begins with the *messageType (m)* field. **Cboe reserves the right to add message types and grow the length** of any message without notice. Participants should develop their decoders to deal with unknown message types and messages that grow beyond the expected length. Messages will only be grown to add additional data to the end of a message.

Data Types

The following field types are used within Cboe One feed. All types are rendered in printable ASCII.

- **Alphanumeric** fields are strings of characters that are not meant for conversion into another data type.
- **Integer** fields are convertible into unsigned or long unsigned integers. They may be signed.
- **Decimal** fields are convertible into floats or doubles. They may be signed.
- **Price** fields are convertible into floats or doubles.
- **Boolean** fields contain either the values `true` or `false`.

Trade Date

Throughout this document, the term "Trade Date" is synonymous with the term "Business Date". The term Trade Date is used within this document to match identically named fields in the CFE FIX and BOE specs.

Sequence Field

The *Sequence* field is used for all Futures TOP messages delivered via Amazon MSK. Examples can be found in TOP Messages on page 125. Messages are sequenced on a per-unit basis.

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Sequence	*s*	Integer	Sequence representing the order that the messages were received by the Kafka producer.

TOP Messages

With the exception of **`Time`** messages, each TOP message reflects the update of the top of book or execution of an order in the system.

Time Message Fields

A **`Time`** message is immediately generated and sent when there is a TOP event for a given clock second. If there is no new sequenced TOP event for a given clock second, then no Time message is sent for that second. All subsequent time offset fields for the same unit will use the new *Time* value as the base until another **`Time`** message is received for the same unit. The *Time* field is the number of seconds relative to midnight Central Time, which is provided in the **`Time Reference`** message. The **`Time`** message also includes the *Epoch Time* field, which is the current time represented as the number of whole seconds since the Epoch (Midnight January 1, 1970).

Take care to only compare messages with a *Time Offset (to)* field to **`Time`** messages from the same partition. Otherwise, because ordering cannot be guaranteed between partitions, the incorrect timestamp could be used.

Table 63. Time Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	TI
Time	*t*	Integer	Number of whole seconds from midnight Central Time.
Epoch Time	*ep*	Integer	Number of whole seconds since the Epoch (Midnight January 1, 1970 UTC).

Unit Clear Message Fields

The **`Unit Clear`** message instructs feed recipients to clear all market snapshots for the CFE book in the unit specified in the **`Sequenced Unit Header`** message header. It would be distributed in rare recovery events such as a data center fail-over. It may also be sent on system startup (after daily restart) when there are no persisted GTCs or GTDs.

Table 64. Unit Clear Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	UC
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp.

Time Reference Message Fields

The **`Time Reference`** message is used to provide a midnight reference point for recipients of the feed. It is sent whenever the system starts up and when the system crosses a midnight boundary. All subsequent **`Time`** messages for the same unit will the use the last *Midnight Reference* until another **`Time Reference`** message is received for that unit. The **`Time Reference`** message includes the *Trade Date*, so most other sequenced messages will not include that information.

`Time Reference` messages will be included in a spin response.

Table 65. Time Reference Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	IR
Midnight Reference	*mr*	Integer	Midnight Central Time reference time for subsequent **`Time`** messages, expressed as number of whole seconds since the Epoch (Midnight January 1, 1970 UTC).
Time	*t*	Integer	Number of whole seconds from midnight Central Time.
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp.
Trade Date	*td*	Alphanumeric	Current Trade Date in format "YYYYMMDD"

Futures Instrument Definition Message Fields

The **`Futures Instrument Definition`** message is sent when the system starts up at the beginning of a trading session or an instrument is created or modified during a trading day. A new sequenced message may be sent for a *Symbol* that does not visibly change any attribute.

If the instrument is a spread, then the message contains one or more repeating groups of leg definitions under the name *I*. There is a limit of 4 leg definitions.

The *Report Symbol* field will contain either the weekly (e.g., VX01) or the monthly (e.g., VX) symbol for any simple futures contract. The *Report Symbol* will always contain the standard futures root symbol (e.g., VX) for all spread instruments.

Table 66. Futures Instrument Definition Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	FD
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp or *Unit Timestamp* in this message if it is non-zero.
Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Unit Timestamp	*ut*	Integer	Unit timestamp expressed as number of whole seconds since the Epoch (Midnight, January 1, 1970 UTC).
Report Symbol	*ps*	Alphanumeric	Symbol for product or underlying security.
Expiration Date	*ed*	Alphanumeric	Expiration Date of Instrument in format "YYYYMMDD".
Contract Size	*cS*	Integer	Contract sizes less than 1 are represented with a 0 value; refer to the product specification for the contract size.
Listing State	*Ls*	Alphanumeric	A = Active I = Inactive T = Test
Price Increment	*pi*	Price	Minimum Price Increment.
Contract Date	*cd*	Alphanumeric	Present for single leg instruments only. Absent for spread instruments. The date that should be used in describing the future's third party symbol and the measurement period of the contract. Set to same value as *Expiration Date* for futures that have a *Contract Date* that does not differ from expire date. Format "YYYYMMDD".
The following fields repeat *Leg Count* times (maximum of 4) for spread instruments.			
Leg Ratio	*ra*	Decimal	Leg ratio (positive for bid-side, negative for ask-side).
Leg Symbol	*sy*	Alphanumeric	Symbol of leg.

Futures Variance Symbol Mapping Message Fields

The **`Futures Variance Symbol Mapping`** message is used to disseminate symbol reference data for S&P 500 Variance Futures (VA Futures) symbols. VA Futures symbol reference data are disseminated with both the **`Futures Instrument Definition`** and **`Futures Variance Symbol Mapping`** messages. The purpose of the **`Futures Variance Symbol`** message is to disseminate product-specific supplemental information for VA Futures symbols (i.e., Accrued Day Variance, Num Final Returns, and Num Elapsed Returns).

Table 67. Futures Variance Symbol Mapping Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	VS
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp or *Unit Timestamp* in this message if it is non-zero.
Unit Timestamp	*ut*	Integer	Unit timestamp expressed as number of whole seconds since the Epoch (Midnight, January 1, 1970 UTC).
Feed Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Futures Symbol	*ps*	Alphanumeric	Twelve character textual definition of the symbol where the first six characters contain the product symbol, left justified, and padded on the right with spaces, and the right most six characters are the expiration date in YYMMDD format.
Accrued Day Variance	*AV*	Decimal	Accrued day variance as of the start of the trading day with twelve decimal place precision.
Num Final Returns	*N*	Integer	Number of S&P 500 Index returns used in the Final Settlement Value calculation.
Num Elapsed Returns	*n*	Integer	Number of elapsed S&P 500 Index returns, including the current day.

Price Limits Message Fields

The **`Price Limits`** message is sent out at the start of the session for products that are subject to price limits per the contract specifications. The **`Price Limits`** message does not signal whether price limits are in effect for that symbol; it simply provides those values for when they are in effect. If multiple **`Price Limits`** messages are received for the same Symbol, the most recent values will override the previous values.

Table 68. Price Limits Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	PL
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp.
Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Upper Price Limit	*ub*	Price	Upper price limit
Lower Price Limit	*lb*	Price	Lower price limit

Market Snapshot Message Fields

A **`Market Snapshot`** message provides a snapshot of the price and size for the bid and ask, last trade price, total number of contracts traded, and the current trading status of a single symbol.

The *Unit Timestamp* field is provided because the timestamp for a **`Market Snapshot`** message is the last time an event occurred on that *Symbol*. Since the Futures market can cross midnight Central Time, the Epoch (midnight, January 1, 1970 UTC) is used as a reference point.

Table 69. Market Snapshot Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	MM
Time Offset	*to*	Integer	Nanosecond offset from *Unit Timestamp* in this message.
Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Unit Timestamp	*ut*	Integer	Last unit timestamp expressed as number of whole seconds since the Epoch (Midnight, January 1, 1970 UTC).
Bid Price	*bP*	Price	Bid price (may be a zero or negative price for some instruments).
Bid Quantity	*bS*	Integer	Number of contracts on the bid side of the inside book (a zero value denotes the *Bid Price* is invalid).
Ask Price	*aP*	Price	Ask price (may be a zero or negative price for some instruments).
Ask Quantity	*aS*	Integer	Number of contracts on the ask side of the inside book (a zero value denotes the *Ask Price* is invalid).
Last Trade Price	*tP*	Price	Price of last execution (this can be zero or negative for some instruments).
Last Trade Size	*tS*	Integer	Number of contracts traded on the last trade (if this value is 0 the *Last Trade Price* is invalid).
Last Trade Condition	*tC*	Alphanumeric	Trade Condition for Last Trade (Space) = Normal trade O = Opening trade S = Spread trade B = Block trade E = ECRP trade X = Trade break
Total Volume	*v*	Integer	Total number of contracts traded on the current business day.
Trading Status	*h*	Alphanumeric	See *Trading Status* field of **`Trading Status`** message.

Single Side Update Message Fields

`Single Side Update` messages provide an updated price and size for a single side of a *Symbol*. The side is denoted by the *Side* field. One **`Single Side Update`** message may reflect one or more updates to the inside book that were processed at the same time.

Table 70. Single Side Update Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	SS
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp.
Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Side	*sd*	Alphanumeric	`B` = Bid side `S` = Ask side
Price	*p*	Price	Price (may be a zero or negative price for some instruments).
Quantity	*sz*	Integer	Number of contracts on the inside book (a zero value denotes the *Price* is invalid).

Two Side Update Message Fields

`Two Side Update` messages provide an updated price and size for both sides of a ***Symbol***. One **`Two Side Update`** message may reflect one or more updates to the inside book that were processed at the same time.

Table 71. Two Side Update Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	TS
Time Offset	*to*	Integer	Nanosecond offset from *Unit Timestamp* in this message.
Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Bid Price	*bP*	Price	Bid price (may be a zero or negative price for some instruments).
Bid Quantity	*bS*	Integer	Number of contracts on the bid side of the inside book (a zero value denotes the *Bid Price* is invalid).
Ask Price	*aP*	Price	Ask price (may be a zero or negative price for some instruments).
Ask Quantity	*aS*	Integer	Number of contracts on the ask side of the inside book (a zero value denotes the *Ask Price* is invalid).

TOP Trade Message Fields

The **TOP Trade** message provides information about executions of orders on the CFE book. **TOP Trade** messages are necessary to calculate CFE execution-based data. **TOP Trade** messages do not alter the book. One or more **Single Side Update** or **Two Side Update** messages will follow a **TOP Trade** message to reflect the updated book (for example, an aggressive order may take out one or more price levels and establish a new level on the opposite side).

Any order may be executed in parts. A complete view of all CFE executions can be built from all **TOP Trade** messages.

The **TOP Trade** message sends the trade price, trade quantity, execution id, and trade condition of a trade as well as the cumulative volume for the business day. A **TOP Trade** message will be sent for each execution, but not every **TOP Trade** message indicates a trade. The Trade Condition value of X (Trade Break) is sent whenever an execution on CFE is broken. Trade breaks are rare and only affect applications that rely upon CFE execution-based data. Trade breaks will contain the Symbol, Quantity, Price, and Execution Id of the original trade. The Total Volume field will be reduced by the number of shares reported in the Quantity field.

Table 72. TOP Trade Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	T
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp.
Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Quantity	*sz*	Integer	Incremental number of contracts executed or corrected (see *Trade Condition*).
Price	*p*	Price	The execution price of the order.
Execution Id	*e*	Alphanumeric	CFE generated day-unique execution identifier of this trade. *Execution Id* is also referenced in the **Trade Break** message.
Total Volume	*v*	Integer	Total number of contracts traded on the current business day (may decrease if the *Trade Condition* field indicates a canceled trade).
Trade Condition	*tc*	Alphanumeric	(Space) = Normal trade O = Opening trade[1] S = Spread trade[1] B = Block trade E = ECRP trade [1]Sent for simple (non-spread) symbols only.

Settlement Message Fields

Sent after the close as part of the end of a Trading Day.

`Settlement` messages are used to provide information concerning indicative, approved, or corrected daily and final settlement prices for CFE products. An indicative daily settlement price (Issue = I) is calculated by the system and sent immediately after an instrument closes trading but before the settlement price is approved. An approved settlement price (Issue = S) is sent once the CFE Trade Desk approves a settlement price for an instrument. If there is an error in the approved settlement price, then it may be re-issued (Issue = R). For symbols that settle each day using VWAP, the system will begin disseminating an intermediate indicative price update (Issue = i) at 2:59:35 (following the first interval of the VWAP calculation) that will be sent every five seconds leading up to the receipt of the indicative daily settlement price (Issue = I).

Table 73. Settlement Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	FS
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp.
Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Trade Date	*td*	Alphanumeric	Trade Date for the settlement in format "YYYYMMDD".
Settlement Price	*sp*	Price	Settlement Price
Issue	*i*	Alphanumeric	i = Periodic Indicative Settlement I = Indicative Settlement S = Initial Settlement R = Re-issued Settlement

End of Day Summary Message Fields

The **`End of Day Summary`** message is sent out right after trading ends for a symbol. No more Market Update messages will follow an **`End of Day Summary`** message for a particular symbol. A value of zero in the *Total Volume* field means that no volume traded on that symbol for the day. The *Total Volume* field reflects all contracts traded during the day. Block and ECRP trades are included in the *Total Volume* field, but they are also reported separately to provide more detail.

The Boolean flag fields provide additional information on how to interpret the *High Price* and *Low Price* fields, especially in instruments that had no volume for the day and/or where 0 is a valid price (e.g., Trade At Settlement products). There are flags that indicate whether or not the *High Price* and *Low Price* fields are valid (*High Price Valid* and *Low Price Valid*). If they are not valid, then there was no High (and/or Low) Price for the day. There are also flags that indicate whether the High Price was set by the highest bid and the Low Price was set by the lowest offer rather than a trade (*High is Bid* and *Low is Offer*).

All **`End of Day Summary`** message values will span the full trading day, including all extended hours trading and all trading segments.

Table 74. End of Day Summary Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	FE
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp.
Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Trade Date	*td*	Alphanumeric	Trade Date for the message in format "YYYYMMDD".
Open Interest	*oi*	Integer	Prior Trade Date Open Interest for this symbol.
High Price	*hP*	Price	The higher of highest bid price and highest trade price for the day. Block and ECRP trades (*Trade Condition* = B or E) do not update *High Price*.
Low Price	*lP*	Price	The lower of lowest offer price and lowest trade price for the day. Block and ECRP trades (*Trade Condition* = B or E) do not update *Low Price*.
Open Price	*oP*	Price	The first trade on the day (in any session) will set the *Open Price* for the day (valid only if *Total Volume* > 0). Block and ECRP trades (*Trade Condition* = B or E) do not update *Open Price*.
Close Price	*cP*	Price	The last trade on the day (in any session) will set the *Close Price* for the day (valid only if *Total Volume* > 0). Block and ECRP trades (*Trade Condition* = B or E) do not update *Close Price*.
Total Volume	*v*	Integer	Total number of contracts traded for the day, including Block and ECRP trades.
Block Volume	*bv*	Integer	Total number of block contracts traded for the day.
ECRP Volume	*ev*	Integer	Total number of contracts traded for the day.
High Price Valid	*hv*	Boolean	Set if *High Price* is a valid value.
High Is Bid	*hb*	Boolean	Set if *High Price* was set by the highest bid (rather than a trade).
Low Price Valid	*lv*	Boolean	Set if *Low Price* is a valid value.
Low Is Offer	*lo*	Boolean	Set if *Low Price* was set by the lowest offer (rather than a trade).

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Open/Close Valid	*ho*	Boolean	Set if both. *Open Price* and *Close Price* fields contain valid values

Trading Status Message Fields

The **`Trading Status`** message is used to indicate the current trading status of a Futures contract. A **`Trading Status`** message will be sent whenever a security's trading status changes. If a **`Trading Status`** message has not been received for a symbol, then the Trading Status for the symbol should be assumed to be S = Suspended. The following summarizes the Trading Status values in the CFE system:

- S = Suspended. A contract is in a suspended state when the associated product is closed and not accepting orders.
- Q = Accepting orders for queuing. Queuing state is used during the Pre-Open for all products. It is also used for spread instruments that may not be tradeable due to Threshold Width.
- T = Trading. Used for both Extended and Regular Hours trading.
- H = Halt state. This state is used for Supervisory Halts initiated by the Trade Desk. Orders are not being accepted in this state.

Table 75. Trading Status Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	TS
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp.
Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Trading Status	*h*	Alphanumeric	S = Suspended Q = Queuing T = Trading H = Halted

Open Interest Message Fields

The **`Open Interest`** message is sent to communicate a symbol's open interest, usually for the prior trading date. This message will be sent when open interest information is made available to CFE and may be sent multiple times if there are changes to the open interest for a symbol. The open interest is also populated in the **`End of Day Summary`** message.

Table 76. Open Interest Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	O
Time Offset	*to*	Integer	Nanosecond offset from last unit timestamp.
Symbol	*sy*	Alphanumeric	Six character, base 62 symbol.
Trade Date	*td*	Alphanumeric	Trade Date for the *Open Interest* data in the format "YYYYMMDD".
Open Interest	*oi*	Integer	Open Interest for this symbol.

End of Session Message Fields

The **`End of Session`** message is sent for each unit when the unit shuts down. No more sequenced messages will be delivered for this unit, but heartbeats from the unit may be received.

Table 77. End of Session Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
Message Type	*m*	Alphanumeric	ES
Timestamp	*to*	Integer	Nanosecond offset from last unit timestamp.

Example Messages

Time Message Example

```
{
 "m" : "TI",
 "t" : 1000,
 "ep" : 1634516200,
 "s" : 20
}
```

Unit Clear Message Example

```
{
 "m" : "UC",
 "to" : 478000000,
 "s" : 478
}
```

Time Reference Message Example


```
{
 "m" : "IR",
 "mr" : 1634533200,
 "t" : 1000,
 "to" : 478000000,
 "td" : "20211019",
 "s" : 47810
}
```

Futures Instrument Definition Single Leg Message Example


```
{
 "m" : "FD",
 "to" : 478000000,
 "sy" : "TESTA",
 "ut" : 1634534200,
 "ps" : "A",
 "ff" : false,
 "ed" : "20211019",
 "cS" : 100,
 "Ls" : "T",
 "pi" : "0.05",
 "cd" : "20211019",
 "s" : 901
}
```

Futures Instrument Definition Multileg Message Example


```
{
 "m" : "FD",
 "to" : 478000000,
 "sy" : "TESTA",
 "ut" : 1634534200,
 "ps" : "A",
 "ff" : false,
 "ed" : "20211019",
 "cS" : 100,
 "Ls" : "T",
 "pi" : "0.05",
 "l" : [
 {
  "ra" : 1,
  "sy" : "TESTB"
 },
 {
  "ra" : -2,
  "sy" : "TESTC"
 },
 {
  "ra" : -3,
  "sy" : "TESTD"
 }
 ],
 "s" : 7050
}
```

Futures Variance Symbol Mapping Message Example


```
{
 "m" : "VS",
 "to" : 478000000,
 "ut" : 1634534200,
 "sy" : "TESTV",
 "ps" : "VA 240621",
 "AV" : 513.523553640453,
 "N" : "505",
 "n" : 490
}
```

Price Limits Message Example


```
{
 "m" : "PL",
 "to" : 478000000,
 "sy" : "TESTA",
 "ub" : "1.00",
 "lb" : "0.50",
 "s" : 78430
}
```

Market Snapshot Message Example


```
{
 "m" : "MM",
 "to" : 478000000,
 "sy" : "TESTA",
 "ut" : 1634534200,
 "bP" : "1.00",
 "bS" : 100,
 "aP" : "2.00",
 "aS" : 200,
 "tP" : "1.50",
 "tS" : 150,
 "tC" : "O",
 "v" : 300,
 "h" : "T",
 "s" : 120030
}
```

Single Side Update Message Example


```
{
 "m" : "SS",
 "to" : 478000000,
 "sy" : "TESTA",
 "sd" : "B",
 "p" : "1.00",
 "sz" : 100,
 "s" : 9014
}
```

Two Side Update Message Example


```
{
 "m" : "TS",
 "to" : 478000000,
 "sy" : "TESTA",
 "bP" : "1.00",
 "bS" : 100,
 "aP" : "2.00",
 "aS" : 200,
 "s" : 40012
}
```

Trade Message Example


```
{
 "m" : "T",
 "to" : 478000000,
 "sy" : "TESTA",
 "sz" : 100,
 "p" : "1.00",
 "e" : "0000EXEC1",
 "v" : 200,
 "tc" : "O",
 "s" : 77812
}
```

Settlement Message Example

```
{
 "m" : "FS",
 "to" : 478000000,
 "sy" : "TESTA",
 "td" : "20211019",
 "sp" : "1.00",
 "i" : "i",
 "s" : 580627
}
```

End of Day Summary Message Example


```
{
 "m" : "FE",
 "to" : 478000000,
 "sy" : "TESTA",
 "td" : "20211019",
 "oi" : 100,
 "hP" : "2.00",
 "lP" : "1.00",
 "oP" : "1.20",
 "cP" : "1.60",
 "v" : 200,
 "bv" : 300,
 "ev" : 400,
 "hv" : true,
 "hb" : true,
 "lv" : true,
 "lo" : true,
 "ho" : true,
 "s" : 500235
}
```

Trading Status Message Example

```
{
 "m" : "TS",
 "to" : 478000000,
 "sy" : "TESTA",
 "h" : "T",
 "s" : 90123
}
```

Open Interest Message Example

```
{
 "m" : "O",
 "to" : 478000000,
 "sy" : "TESTA",
 "td" : "20211019",
 "oi" : 100,
 "s" : 2187
}
```

End of Session Message Example

```
{
 "m" : "ES",
 "to" : 478000000,
 "s" : 4598
}
```

Cboe Global Indices Feed

Overview

Cboe members may use the Cboe Global Indices Feed (formerly CSMI) to receive real-time index data.

Index messages contain the values associated with a calculated index. For some indices, a bid and ask value may also be calculated which is like the index value but is calculated from bid and ask prices instead of last sale prices. Index values are benchmark values upon which tradable products may be based, but an index itself is not tradable.

Message Distribution

The Kafka topic for Cboe Global Indices is partitioned by symbol - maintaining order within a given symbol to customers.

Index calculation groups, or channels, are distributed on separate topics: CCCY, Cboe Global Indices MAIN, FTSE, INAV, MSCI, Morningstar, Morningstar Custom, and CGIDF. Cboe reserves the right to make changes to these groups as needed.

Message Format

The Cboe Global Indices feed is comprised of a series of Kafka JSON messages. Each message has a messageType (msgType) property that indicates the type of message. Fields are properties that conform to the standard JSON pattern field name: field value.

Data Types

Within the JSON format the following format is used for each data type.

- **Numeric** fields are a string of ASCII digits that can be converted to an integer.
- **Number** fields represented in base 10 and can be converted to a floating point number
- **Alpha** and **Alphanumeric** represent one or more printable ASCII characters.

Cboe Global Indices Messages

Index Value Message Fields

`Index Value Update` messages contain values associated with calculated index. Each message will contain one or more Index Value Blocks.

Table 78. Index Value Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
sendTime	*sendTime*	Numeric	Epoch time of value was published from system
messageType	*msgType*	"X"	X = index value update
sequence number	*seq*	Numeric	Sequence identifier from Cboe Global Indices (Note: seq on a given topic will be increasing but not guaranteed to be without gaps). Reset is expected on restart or failover.
symbol	*sym*	Alpha	Index Symbol
IndexValueBlocks	*values*	JSON List	List of Index Value Block JSON objects, defined below

Table 79. Index Value Block Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
type	*type*	Numeric	0 = bid 1 = ask 3 = index value
value	*value*	Number	Value associated with given index data point

Contributor Value Message Fields

`Contributor Value` messages contain values associated with an index or symbol received from the Cboe Contributor API. Each message will contain one or more Index Value Blocks.

Table 80. Contributor Value Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
sendTime	*sendTime*	Numeric	Epoch time value was published from system
transactionTime	*transTime*	Numeric	Epoch time the transaction occurred
messageType	*msgType*	"C"	C = contributor value update
sequence number	*seq*	Numeric	Sequence identifier from Cboe Global Indices (Note: *seq* values will be increasing but not guaranteed to be without gaps). Reset is expected on restart or failover.
symbol	*sym*	Alpha	Index Symbol
IndexValueBlocks	*values*	JSON List	List of Index Value Block JSON objects, defined below

Table 81. Index Value Block Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
type	*type*	Numeric	0 = bid 1 = ask 3 = index value
value	*value*	Number	Value associated with given index data point

Index Summary Message Fields

`Index Summary` messages contain values associated with start or end of day values. Effective date is included for reference. Bid and ask fields are not required. Each message will contain one or more Index Value Blocks.

Table 82. Index Summary Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
sendTime	*sendTime*	Numeric	Epoch time of value was published from system
transactionTime	*transTime*	Numeric	Epoch time the transaction occurred
messageType	*msgType*	"S"	S = index summary update
summaryType	*summaryType*	"1"	Type of summary 1 = SOD 2 = EOD
sequence number	*seq*	Numeric	Sequence identifier from Cboe Global Indices (Note: *seq* values will be increasing but not guaranteed to be without gaps). Reset is expected on restart or failover.
symbol	*sym*	Alpha	Index Symbol
IndexValueBlocks	*values*	JSON List	List of Index Value Block JSON objects, defined below

Table 83. Index Value Block Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
type	*type*	Numeric	0 = bid 1 = ask 3 = index value
value	*value*	Number	Value associated with given index data point

Index Definition Message Fields

`Index Definition` messages contain values associated with information describing an index such as an index's current status (active, inactive, or deleted), channel name, current trading date, data source, and description.

Table 84. Index Definition Message Fields

FIELD NAME	JSON	VALUE	DESCRIPTION
sendTime	*sendTime*	Numeric	Epoch time message was published from system.
messageType	*msgType*	"D"	D = definition message update.
sequence number	*seq*	Numeric	Sequence identifier from Cboe Global Indices (Note: seq values will be increasing but not guaranteed to be without gaps). Reset is expected on restart or failover.
symbol	*sym*	Alpha	External index symbol.
current record number	*currRec*	Numeric	Clients should begin capturing data beginning at 1.
total number of records	*totRecs*	Numeric	Total number of updates within loop.
description	*desc*	Alpha	Description of the index.
date	*date*	Alpha	Current business date for a given index symbol.
status	*status*	Numeric	Status of Index. 1 = Active - loaded and a business day 2 = Inactive - loaded and not a business day 3 = Deleted - no longer distributed (removed from the feed after '30' days
agent classification	*agentCode*	Alpha	Originating source of the index: <blank> = Not Specified CO = Cboe CC = Cboe Custom MC = Morningstar Customer MS = Morningstar TP = Third-party
channel	*channels*	JSON List	Sequence Group of the following types: CCCY = Cryptocurrency CSMI = Main CGI = European Markets INAV = Net Asset Values FTSE = Russell Indices MSCI = Morgan Stanley MSTAR = Morningstar

Cboe Global Indices Example Messages

The following are examples of each message type that can be sent on Cboe Global Indices.

Index Value Message Example (with bid / ask)


```
{
"sendTime": 1635171838052,
"msgType": "X",
"seq": 95742,
"sym": "SPX",
"values": [
 {
 "type": "3",
 "value": 4551.11
 },
 {
 "type": "0",
 "value": 4549.77
 },
 {
 "type": "1",
 "value": 4552.14
 }
]
}
```

Index Value Message Example (only last sale)


```
{
"sendTime": 1635171646325,
"msgType": "X",
"seq": 91797,
"sym": "VIX",
"values": [
 {
 "type": "3",
 "value": 15.73
 }
]
}
```

Contributor Message Example (only last sale)


```
{
"msgType": "C",
"sendTime": 1646069383825,
"transTime": 1646069382825,
"seq": 100,
"sym": "TSTSYM",
"values": [
 {
 "type": "3",
 "value": 1.23
 }
]
}
```

Index Summary Message Example (only last sale)

```
{
"msgType": "S",
"sendTime": 1646069113466,
"transTime": 1646069112466,
"effectiveDate": "20220228",
"summaryType": "1", // SOD
"seq": 100,
"sym": "TSTSYM",
"values": [
 {
 "type": "3",
 "value": 1.23
 }
]
}
```

Index Definition Message Example


```
{
"sendTime": 1443707509082,
"msgType": "D",
"seq": 7565,
"sym": "CLLZ",
"currRec": 11,
"totRecs": 500,
"desc": "Collar Zero Cost Index Calculation",
"date": "2023-02-24",
"status": 1,
"agentCode": "CO",
"channels": [
"CSMI"
]
}
```

Cboe Europe Last Sale Feed for BXE, CXE, DXE (EU)

The Last Sale feed supplements the Top of Book streams, providing a detailed, non-compressed stream of distinct trades. All applicable trade types are included (dark book executions, exchange trade reports, periodic auctions, etc.), with full MMT type information.

Last Sale Message

Trade Message Fields

Table 85. Trade Message Fields

FIELD NAME	JSON	VALUE/TYPE	DESCRIPTION
messageType	*m*	"T"	**`Trade`** message
timestamp	*t*	Integer	Milliseconds since midnight.
symbol	*sy*	Alphanumeric	Relevant symbol.
executionId	*e*	Alphanumeric	Market center specific execution identifier of this Execution. *Execution ID* is also referenced in the Trade Break message.
tradePrice	*tp*	Price	Last trade price.
tradeSize	*ts*	Integer	Last trade quantity.
tradeTimestamp	*tt*	Integer	The time the trade occurred on the specified Market Center.
flags	*f*	Alphanumeric	14-character representation of MMT v4.1 flags.
venue	*v*	Alphanumeric	ISO 10383 segment MIC indicating where the trade executed. E.g., BATE, CHID, LISZ, etc.
sequence	*s*	Integer	Sequence representing the order that the messages were received by the Kafka producer.

Market Model Typology (MMT)

The MMT (Market Model Typology) Initiative is a collaborative effort established by a broad range of industry participants (trading/reporting venues, data vendors and buy/sell side participants). The initiative is committed to achieving a practical and common solution for standards on post-trade data across all asset classes subject to MiFID II. The initiative unites a variety of industry participants in the basic belief that we can and should act without any further delay to improve the consistency and comparability of data from different sources. MMT support has been upgraded to v4.1. The full list of supported values can be found in the Cboe Europe Multicast PITCH Specification. For further information and resources please refer to the FIX trading community webpage.

Trade Breaks/Cancellations

The **`Trade`** message is also used to convey when a trade of any type (on-book execution, Exchange Trade Report, etc.) is cancelled. The trade is repeated with the same ***executionId***, but with the appropriate cancellation flag set within the MMT flags field. This applies equally to on-book executions busted by Trade Desk staff, and to ETRs which may be cancelled by Participants themselves.

Trade Amendments

Reporting Participants may amend ETRs, and the Cboe Trade Desk may amend any trade. In either case, the feed will first send a message cancelling the initial version of the trade (with the MMT cancel flag set), followed immediately by a second message re-stating the trade with the new details. Both the cancel and the amendment messages will have the ***executionId*** of the original message.

Last Sale Example Messages

Trade Message Example


```
{
    "m": "T",
    "t": 36000123,
    "sy": "VODl",
    "e": "00WY30000002",
    "tp": "320.20",
    "ts": 150,
    "tt": "
2022-05-30 09:41:07.717202",
    "f": "1O-------PH---",
    "v": "CEUX",
    "s": 123
}
```

Trade Break/Cancellation Message Example


```
{
    "m": "T",
    "t": 36000123,
    "sy": "VODl",
    "e": "00WY30000002",
    "tp": "320.20",
    "ts": 150,
    "tt": "
2022-05-30 09:41:07.717202",
    "f": "1O---C---PH---",
    "v": "CEUX",
    "s": 112
}
```

Trade Amendments Message Example

In this example, an ETR has been amended, changing size from 150 to 160.


```
{
    "m": "T",
    "t": 36000123,
    "sy": "VODl",
    "e": "00WY30000002",
    "tp": "320.20",
    "ts": 150,
    "tt": "
2022-05-30 09:41:07.717202",
    "f": "1O---C---PH---",
    "v": "CEUX",
    "s": 145
}
{
    "m": "T",
    "t": 36000123,
    "sy": "VODl",
    "e": "00WY30000002",
    "tp": "320.20",
    "ts": 160,
    "tt": "
2022-05-30 09:41:07.717202",
    "f": "1O---A---PH---",
    "v": "CEUX",
    "s": 146
}
```

References

Symbology

For more information on Cboe Symbology, please refer to Cboe Symbology Reference.

Support

Please direct questions or comments to the corresponding trade desk:

NAME	CONTACT INFO
EU TOP and Last Sale Feeds (BXE, CXE, DXE)	tradedeskeu@cboe.com
AU TOP and PITCH Feeds	tradedeskau@cboe.com
JP PITCH Feeds	tradedeskjp@cboe.com
US TOP (BZX and EDGX) and Cboe One Feed	tradedesk@cboe.com
US Futures TOP Feed	cfetradedesk@cboe.com
Cboe Streaming Market Index Feed	indexsupport@cboe.com
Canadian Cboe One Feed	tradedeskca@cboe.com

Revision History

DOCUMENT VERSION	DATE	DESCRIPTION
1.0.0	08/17/21	Initial version.
1.1.0	10/27/21	Added Futures TOP Feed and CSMI sections.
1.1.1	11/08/21	Updated links in the Kafka Configuration section.
1.1.2	02/28/22	Added notes to support Contributor API protocol (effective 05/01/22).
1.1.3	06/16/22	Added support for Cboe Australia Equities.
1.1.4	06/24/22	Addition of EU TOP and Last Sale feed for BXE, CXE and DXE.
1.1.5	07/07/22	Added contact information to Support section.
1.1.6	07/21/22	Added sequence field to Cboe Europe Last Sale trade message.
1.1.7	09/06/22	Updated Cboe Global Indices Feed (formerly CSMI) branding and added Morningstar Channels. Added Cboe One Canada Feed in section 4 (effective 09/12/22). Updated *Symbol* JSON to 'sy' in sections 5, 6, and 7. Removed *Bid is Valid* and *Ask is Valid* from AU Equities TOP sections 6.6.3, 6.6.4, and 6.6.5. Updated *Trade Quantity* JSON to 'sz' in sections 6 and 7. Updated *Bid Price*, *Bid Quantity*, *Ask Price*, and *Ask Quantity* JSONs in section 6 to 'bP', 'bS', 'aP', and 'aS', respectively. Linked to the Cboe Global Cloud Setup Guide in section 2.
1.1.8	10/10/22	Updated AU Equities PITCH Feed *Message Type* value to 'x' for **`Trade Break`** messages.
1.1.9	10/24/22	Updated *Symbol* JSON to 's' in section 5.5.
1.1.10	12/07/22	Updated *timestamp* description in section 5.5. to include UTC time zone.
1.1.11	01/26/23	Updated *Trade Designation* description in sections 6.6.6 and 7.6.8 to include values "B" and "I" for BIDS MIC codes.
1.1.12	02/28/23	Added cgi-idx-ape1, cgi-idx-use1, and cgi-idx-euw2 to Kafka Topics.
1.1.13	03/13/23	Added definition-idx-ape1, definition-idx-euw2, and definition-idx-use1 to Kafka Topics (effective 03/27/23). Added new **`Index Definition`** Message sections to section 9 (effective 03/27/23).
1.1.14	04/10/23	*nationalVolume* in Canada is the sum of two values: the current Cboe Cumulative Executed Volume and the 15-minute delayed executed volume of other Canadian exchanges as reported by TMX IP. Added *marketCenter* = T, 'V', 'S', 'c', 'r', and 't'. Updated `Index Definition` channel description to CGI=European Markets.
1.1.15	04/24/23	Added AU and JP links to Protocol section. Updated *Message Type* description in section 7.6.3. to "Add Order Message". Added CXJA PITCH and CXJS PITCH recommended bandwidths. Added Cboe One Canada Premium and Cboe One Canada Summary recommended bandwidths. Added section 8: Cboe JP Equities PITCH Feed. Added Tokyo data feed Kafka topics (ap-northeast-1). Added CXJA and CXJS broker information. Added "Cboe JP Equities PITCH Feed" protocol section. Added CXJ contact information.
1.1.16	10/17/23	MMT support will be upgraded to v4.1 (effective 01/01/24).

DOCUMENT VERSION	DATE	DESCRIPTION
1.1.17	03/13/24	Added new Agent Classifications MC and MS (effective 06/23/24).
1.1.18	05/27/24	Cboe AU Equities TOP Feed - updates to accommodate Auctions on CXA Symbols: Trading Status message - updated Trade message - updated Auction Update message - new Auction Summary message - new. Cboe AU Equities PITCH Feed - updates to accommodate Auctions on CXA Symbols: Trading Status message - updated Order Executed message - updated Auction Update message - new Auction Summary message - new All CXA Auction updates temporarily highlighted in red for clarity. (Effective date TBC - late Q4, 2024)
1.1.19	06/28/24	Added **`Futures Variance Symbol Mapping`** to section 9.4, added new message structure for **`Futures Instrument Definition`** (replacing the current message structure), added **`Futures Variance Symbol Mapping`** message and example (effective 09/23/24).
1.1.20	08/01/24	Updated timestamp field in section 5.5 TOP Messages: "Milliseconds since midnight London."
1.1.21	12/11/24	Updated description in Futures Variance Symbol Mapping Message Fields on page 129. Updated topics in Region: Sydney (ap-southeast-2) on page 16. Updated Best Quote Update Message Fields on page 26 *messageType* to Q. Added Region: Singapore (ap-southeast-1) to Data Feed Kafka Topics (by Region) on page 11.
1.1.22	01/15/25	Updated with Cboe Titanium branding. Updated Cboe One Feed section to indicate the Cboe US Equities topics have 36 partitions, while the Cboe One Canada topics have 4 partitions. Updated **`Trade Message Fields`** *Trade Type* values to include O=Opening Auction Trade/Intra-day Auction Trade , C=Closing Auction Trade, and H=Halt Auction (Reopening). Updated **`Order Executed Message Fields`** *Execution Type* (Et) to *Trade Type* (Tt).
1.1.23	02/10/25	Added *flags* field to AU Equities TOP and PITCH feeds in the **`Trade`** message.
1.1.24	02/28/25	For CBOE AU Equities PITCH and TOP Feeds, updated *Indicative Price*, *Price*, and *Value* values from Numeric to Price for **`Auction Update`**, **`Auction Summary`**, and **`Calculated Value`** messages, and included Bit Field in the list of data types.
1.1.25	03/21/25	Updated **`Trade Message Fields`** *Trade Designation* values to remove B = CXAB (BIDS Block Size) and I = CXAI (BIDS Price Improved) from the Cboe AU Equities TOP and PITCH Feeds.
1.1.26	03/25/25	Updated CXJA and CXJS partitions per topic to 4 in Broker Information on page 20.
1.1.27	06/11/25	Added *haltStatus* = F, M, O, P, and X to the **`Trading Status`** messages (NEO Only) (effective 08/11/25).

DOCUMENT VERSION	DATE	DESCRIPTION
1.1.28	06/23/25	Updated *haltStatus* to *tradingStatus* on **`Trading Status`** messages.
1.1.29	07/31/25	Updated *Contract Size* description to indicate that contract sizes less than 1 are represented with a 0 value; refer to the product specification for the contract size.