



CGIF Specification

Version 1.4.9

August 16, 2023

This content is owned or licensed by Cboe Global Markets, Inc. or its affiliates (“Cboe”) and protected by copyright under U.S. and international copyright laws. Other than for internal business purposes, you may not copy, reproduce, distribute, publish, display, perform, modify, create derivative works, transmit, or in any way exploit the content, sell or offer it for sale, use the content to construct any kind of database, or alter or remove any copyright or other notice from copies of the content.

Table of Contents

Reference Tables.....	3
1 Introduction	5
1.1 System Overview	5
1.2 Hours of Operation.....	5
2 Data Feed and Message Overview	6
2.1 Data Feed Overview.....	6
2.2 Message Overview	6
3 Message Templates, Field Data Types and Data Encoding.....	8
3.1 Message Templates	8
3.2 Template IDs.....	9
3.3 Field Data Types and Data Encoding	9
4 Packet and Message Header Format	12
4.1 Packet Header	12
4.2 Message Header.....	13
5 Messages	15
5.1 Index Value – Template ID 22	15
5.2 Heartbeat Message (Line Integrity Message) – Template ID 16	16
5.3 Contributor Value Message – Template ID 23.....	16
5.4 Index Summary – Template ID 24.....	17
5.5 Index Value With Status – Template ID 25	18
5.6 Index Definition Message - Template ID 26.....	19
6 Appendix A – Multicast Group and Port Information.....	21
7 Appendix B – Examples.....	23
7.1 Understanding the Hex Data Diagrams	23
7.2 Packet Header Example	23
7.3 Index Example	23
7.4 Heartbeat Message.....	24

Reference Tables

1 - Example of an XML based template.....	8
2 – Templates and their IDs.....	9
3 - Packet Format	12
4 - Packet Header.....	12
5 - Message Format.....	13
6 - Message Header.....	13
7 –Message Types	13
8 – Index Value Message Structure	15
9 – Index Value Template	16
10 – MD Entry Type	16
11 - Heartbeat Message Structure	16
12 - Heartbeat Template	16

Examples of Data Transmissions

1 - Packet Header Hex Dump	23
2 - Packet Header Decoded	23
3 - Heartbeat Hex Dump.....	24
4 - Heartbeat Decoded	24

1 Introduction

The CBOE Streaming Market publishes index data using the message format defined in this document. Data is transmitted using the IP Multicast network protocol. To connect to the CFN network, refer to the [Cboe Futures Exchange Connectivity Manual](#).

1.1 System Overview

CBOE Streaming Market distributes index values. A **feed** is a set of one or more data channels. A **channel** consists of two Multicast groups in a primary/secondary architecture where the data is duplicated on the two Multicast groups for redundancy.

Communication is one way only with no mechanism for retransmission. Messages are encoded using a mix of ASCII characters and binary data in the format defined in this document. The templates and message structures defined in this document are static and will not change over the course of the trading day, nor even in most software releases. Clients can expect sufficient advanced notice on any changes to these templates or message structures.

1.2 Hours of Operation

Messages will be published during these hours.

Normal market hours are as follows (CT):

Channel	Dissemination Time	Connectivity/Heartbeating
Cboe Global Indices Feed Main	2:00 a.m. – 7:15 p.m. (Monday – Friday)	
MSCI	8:30 a.m. – 7:15 p.m. (Monday – Friday)	
FTSE		
MSTAR	24x6 (Sunday - Friday)	
MSTARC		
CCCY	24x7	
CGI	2:00 a.m. – 10:30 a.m. (Monday – Friday)	
Cboe Global Indices Definition Feed	24x7 with periodic dissemination on a 15-minute interval maximum*	24x7

*Any changes to the 15-minute maximum dissemination interval will be communicated in a future notice.

2 Data Feed and Message Overview

2.1 Data Feed Overview

The feed consists of one **data** channel. General characteristics of the feed include the following:

- The channel is duplicated and sent to 2 different multicast groups and ports over 2 networks in a primary / secondary configuration. Data sent to the primary and secondary multicast groups for each channel is identical.
- There are no retransmissions. Index values are computed and transmitted at regular intervals.
- A sequence number is sent for each message. This can be used to identify missed messages.
- Messages are placed into blocks (packets) for delivery which allows for multiple messages per block. The maximum block size is 1000 bytes.
- The message structures, field names and field values are based as much as possible on the FIX 5.0 SP2 standard. However, messages are encoded using a proprietary ASCII + binary format, and FIX tags are not transmitted in the data stream. The FIX format was used for the convenience of those familiar with the FIX standard, so messages are defined in terms of FIX field names and FIX tags. Some user-defined fields were necessary for those fields not in the FIX specification, and some modifications to standard FIX fields are implemented for efficiency reasons.

2.2 Message Overview

The following types of messages are transmitted over the feed:

2.2.1 Index Value

Index value 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.

Index values are not sent using classKey and security IDs like other CSM feeds. Index values do not have a security definition associated with them, instead the index symbol is sent in every index value message.

2.2.2 Heartbeats

Heartbeat/line integrity messages are transmitted every five seconds. These messages may be used to determine a channel is working during times when market data is not transmitted on the feed (such as pre-market or post-market times).

2.2.3 Contributor Value

Contributor Value contains values with associated times for a given symbol. For some symbols, the bid and ask value may also be populated.

2.2.4 Index Summary

Index Summary messages are used in conjunction with the Cboe Contributor API specification and provide either the open or close details and associated effective date for a given index. Bid and ask values may also be populated but are not required.

Index Summary messages are not required and should not be assumed to be provided with Index Value messages.

2.2.5 Index Value with Status

Index Value With Status messages contain the values associated with a calculated index. Index values are benchmark values upon which tradable products may be based, but an index itself is not tradable. These index values contain a status describing if the value is indicative or not. Valid entry types also include close and settlement markers within the message.

2.2.6 Index Definition

The Index Definition message published in the Cboe Global Indices Definition Feed (“CGIDF”) contains index information including an index’s current status (active, inactive, or deleted), channel name, current trading date, data source, and description.

Index Definition messages published through the CGIDF can be used optionally in support of and in conjunction with the information provided across all other Cboe Global Indices Feed Channels.

3 Message Templates, Field Data Types and Data Encoding

3.1 Message Templates

Messages for the CBOE Streaming Market are described in this document in tabular text format and as an XML template. Templates define the content and characteristics of the messages to be encoded or decoded. Firms are encouraged to write software capable of using the XML templates to decode data from the CBOE Streaming Market.

XML templates are used to specify the structure, data types, field names, and FIX tags of a message:

1 - Example of an XML based template

<template name="MDIncRefresh" id = "0">	Start of new template
<string name="MessageType" id="35" byteLength="1" value="X" />	Defines a String Data Type Field, The id which represents the fix Tag is not transferred on the wire.
<uInt32 name="MsgSeqNum" id="34" />	
<sequence name="MDEntries">	Defines the start of a repeating group
<length name="NoMDEntries" id ="268"/>	Length of repeating group
<string name="Symbol" id="55"/>	
<uInt32 name="Quantity" id="53" />	
</sequence>	End repeating group
</template>	End template

3.2 Template IDs

Each message structure is defined with a unique template ID. A template ID is a binary integer value stored as the first byte of every message identifying the message structure.

Template IDs are assigned from a common pool for the CBOE Streaming Market, so there may be gaps in numbering when either specification is updated. Numbers are assigned sequentially as they are needed for new message structures and old template ids are retired when new versions of the feed are launched.

The template ids for the CBOE Streaming Market are as follows:

2 – Templates and their IDs

Template Name	Template ID	Assigned in Version
Heartbeat	16	1.0
Index Value	22	1.0
Contributor Value	23	1.3.4
Index Summary	24	1.3.4
Index Value with Status	25	1.4.5
Index Definition	26	1.4.5

3.3 Field Data Types and Data Encoding

Fields defined in messages for the CBOE Streaming Market will have one of the following data types and methods of encoding:

3.3.1 STRING Field

Strings are ASCII character arrays or single-byte characters. There are two types of string fields which are encoded differently:

Single Byte String

If the *byteLength* attribute of a string field is defined as “1”, for example:

```
<string name="MessageType" id="35" byteLength="1"/>
```

it is a single byte string, which is encoded with a single ASCII character.

Character Array String

If the string has a *byteLength* attribute greater than 1 or the *byteLength* attribute is absent, the string is encoded with an unsigned binary byte indicating the actual length of the string, followed by the string's characters:

Length of String (1 byte)	String Characters
---------------------------	-------------------

When the *byteLength* attribute is present, the string is of fixed length equal to *byteLength* but is still encoded with a length byte preceding the characters of the string. The length byte must still be used to decode the string.

When the *byteLength* attribute is absent, the string is encoded with a variable length. The *Comments* field should be used to indicate the field's maximum length.

For example, the *EffectiveBusinessDate* field with *byteLength="8"* would have the string “20230815” encoded as:

Binary value 8, then the characters “20230815”.

3.3.2 SYMBOL Field

Symbol field is encoded as a string field and has a maximum length of 12 characters.

3.3.3 INTEGER and LENGTH Fields

Integer and Length fields are big endian binary encodings of numeric values. The *byteLength* attribute in a template field definition can act as a modifier to restrict the number of bytes used to encode the integer value. Unsigned integers are encoded as zero or positive-only values. The top-most bit is part of the magnitude of the value. Signed integers are encoded as two’s-complement binary values with the top-most bit as the sign bit. Length fields are unsigned integer values used to indicate the length of a Sequence field.

There are several types of integer or length fields:

Integer Field Type	Byte Length	Encoding	Example
ulInt32	1	8 bit unsigned integer	<ulInt32 name="MDPriceLevel" id="1023" byteLength="1"/>
ulInt32	4 or omitted	32 bit unsigned integer	<ulInt32 name="MDEntrySize" id="271" byteLength="4"/>
ulInt64	8 or omitted	64 bit unsigned integer	<ulInt64 name="SendingTime" id="52"/>
int32	1	8 bit signed integer	<int32 name="MDPriceLevel" id="1023" byteLength="1"/>
int32	4 or omitted	32 bit signed integer	<int32 name="MDEntrySize" id="271"/>
int64	8 or omitted	64 bit signed integer	<int64 name="SendingTime" id="52" byteLength="8"/>
length	1	8 bit unsigned integer	<length name="NoMDEntries" id="268" byteLength="1"/>

The table below shows the min and max values for different integer data types.

Type	Min	Max
ulInt32 with byteLength="1"	0	255
ulInt32	0	4,294,967,295
ulInt64	0	18,446,744,073,709,551,615
int32 with byteLength="1"	-128	127
int32	-2,147,483,648	2,147,483,647
int64	- 9,223,372,036,854,775,808	9,223,372,036,854,775,807
length with byteLength="1"	0	255

3.3.4 DECIMAL Field

A decimal field is used to represent a floating point number as exponent and mantissa. The exponent is a signed 8 bit integer used to express precision and the mantissa is a signed 32 bit integer used to express the value. The numerical value is obtained by multiplying the mantissa with the base-10 power of the exponent expressed as: $\text{number} = \text{mantissa} * 10^{\text{exp}}$. The exponent and mantissa is decoded as a single, composite field.

Decimal fields are 5 bytes in length. The first byte is the exponent, and the remaining 4 bytes are the mantissa. For example, the number 0.90 is encoded as FE0000005A.

FE (exponent) == -2, 0000005A (mantissa) == 90, value == $90 * 10^{-2} == 90 * 0.01 == 0.90$

3.3.5 SEQUENCE Field

A sequence is a repeating group of fields. A length field encoded as an unsigned int immediately precedes the fields contained in the sequence. The length field is defined in a template with a special attribute of “<length”, and it can be modified with *byteLength* attribute. If *byteLength*==”1”, the encoded length field is a single 8-bit unsigned byte. All sequences transmitted to the CBOE Streaming Market use a single 8-bit unsigned length and can be no longer than 255 entries.

Sequences are encoded as follows:

Length field	Group#1 Field #1	Group#1 Field#2	...	Group#1 Field#N	Group#2 Field#1	Group#2 Field#2	...	Group#2 Field#N	...
--------------	------------------	-----------------	-----	-----------------	-----------------	-----------------	-----	-----------------	-----

Here is an example sequence with 2 MDEntries elements.

Field	Length in Bytes	Value	Comments
NoMDEntries	1	2	Length of MDEntries Sequence
MDEntryType	1	'0'	Bid entry
MDEntryPx	5	FE0000000A	FE == -2 exponent, 0000000A == 10 mantissa, value == 0.10
MDEntryType	1	'1'	Ask entry
MDEntryPx	4	FE00000010	FE == -2 exponent, 00000010 == 16 mantissa, value == 0.16

4 Packet and Message Header Format

All messages are sent in Multicast packets. Each packet consists of a packet header and one or more messages.

3 - Packet Format

Packet (a.k.a. Block)					
Packet Header					Contents
Version	Length	Sending Time	Number of messages	First Msg Seq #	Messages...

4.1 Packet Header

Each packet has a packet header appearing once at the beginning of the packet. The packet header has the following structure:

4 - Packet Header

Field Name	Type	Length (Bytes)	Comments
Version	ulInt32	1	The version associated with the contents and format of this header. Currently, this will be a constant value of 1.
Length	ulInt32	2	Length of the packet including this length field and the version. Note this is a 2 byte length.
Sending Time	ulInt64	8	The time this packet was sent. It applies to all messages in this packet.
Number of messages	ulInt32	1	The number of messages in this packet.
First Msg Seq #	ulInt32	4	The sequence number on the first message in this packet.

The version of a packet indicates the format of the packet. This may be incremented in future releases to indicate a change in the format of the packet. Initially, it is set to the number 1.

The Packet Length is encoded as a 2 byte (16 bit) unsigned integer including the length of the version, the 2 byte Packet Length itself, and the remainder of the packet.

The Sending Time is the time the CBOE Streaming Market application published the packet on the feed. The sending time is the millisecond timestamp from midnight, January 1, 1970 UTC.

The “First Msg Seq #” is the sequence number of the first message of this packet, and the “Number of Messages” indicates the total number of messages contained in the packet.

For verification of data at the channel level, one could compute the expected “first msg seq #” of the next packet by adding the number of messages to the current packet’s “first msg seq #”.

4.2 Message Header

A packet contains multiple messages. Each message is preceded by a message header common to all messages.

5 - Message Format

Message						
Message Header			Contents			
Template defined fields						
Length	Template ID	Msg Type	Msg Seq #	Field #1	...	Field #N

6 - Message Header

Field ID	Field Name	Type	Length (Bytes)	Comments
	Message Length	uInt32	2	The length of this message including the 2 bytes for this length field.
	Template ID	uInt32	1	The Template ID is for decoding the message. See table: <i>2 - Templates and their IDs</i>
35	MessageType	String	1	See table: <i>7 -Message Types</i>
34	MsgSeqNum	uInt32	4	Sequence Number

The Message Length is encoded as a 2 byte (16 bit) unsigned integer including itself, the remainder of the message, and all Message Header fields.

The Template ID defines the specific Structure of the message.

The Message Type defines the market data message type compliant to the FIX standard.

The message sequence number is a consecutively increasing number from the previous message. The first message in a packet will start with 1 number greater than the last message in the previous packet

7 -Message Types

Note: Per FIX specifications, message types can be shared, therefore, body of message is determined based on template_id provided

X	Contributor Value Index Summary Index Value Index Value with Status
D	Index Definition Message
0	Heartbeat

4.2.1 Message Sequence Numbers

Every packet has a “first” sequence number in the header. This is the number associated with the first message in the packet. Each subsequent message in the packet has a sequence number one greater than the previous message. The next packet will have a starting sequence number one more than the last message in the previous packet except for the start of the trading day’s session and in the event of a Cboe system failure. When the session is started the sequence number will be reset to 1. During a Cboe system failure the sequence number can reset to a lower value than was previously seen prior to the failure. In either case, the sequence number will again increase by one for each message.

Each channel of a feed has its own sequence number associated with it starting with sequence number 1. Verification of message sequence numbering must be done for each individual channel.

Firms must ensure that the sequence numbers maintain continuity. Any deviation from an expected sequence number must be considered as an error condition. Firms are required to take appropriate recovery action any time an unexpected sequence number is detected.

4.2.2 Recovery from Unexpected Message Sequence Numbers

Each message sent on a channel causes its *MsgSeqNumber* to increment by one. To detect missing data at the channel-level, compare each incoming *MsgSeqNumber* with the last received *MsgSeqNumber*+ 1. If the incoming *MsgSeqNumber* is not equal to the (last received *MsgSeqNumber* + 1), data is missing from the channel.

Regardless of whether missing data is detected or not, the *MsgSeqNumber* of the incoming message should be stored associated with the channel so subsequent missing data can be detected.

When missing data is detected at the channel-level, all market data for products received over that channel should be treated as “suspect”, meaning their market data may be incorrect. At the time missing data is detected on a channel, there is no way to know which index’s data is missing, therefore all indexes received from that channel must be treated as though the market data for those indexes may be incorrect.

Indexes marked as suspect or possibly incorrect should remain in that state until an index value message for that index is received. Use it to update the suspect index’s market data and mark the index’s market data as no longer suspect.

5 Messages

5.1 Index Value – Template ID 22

Index value messages contain the values associated with a calculated index. They are transmitted when an index is calculated.

Market data associated with the index are contained in MDEntries with an MDEntryType indicating the kind of market data. The Index Value message may include any of the following MDEntryTypes (See table 10 – MD Entry Type).

- Index Value
- Bid
- Ask

At least one MDEntry with MDEntryType == Index Value will always be present in the Index Value message. For indices where a Bid and Ask index value is calculated, MDEntryTypes Bid and Ask will be present. For indices where Bid and Ask is not calculated, MDEntryTypes for Bid and Ask are omitted.

8 – Index Value Message Structure

Field ID	Field Name	Type	Length (Bytes)	Comments
	Standard Header			See table: <i>7 –Message Types</i> , MessageType = “X”
55	Symbol	symbol		Index Symbol/iNAV Symbol
268	NoMDEntries	length	1	Number of MDEntries in this message. Will not exceed 255
Repeating Group - Fields Repeat NoMDEntries times				
269	MDEntryType	single byte string	1	Entry Type. See table: 10 – MD Entry Type
270	MDEntryPx	decimal	5	Index value associated with MDEntryType

9 – Index Value Template

```
<template name="IndexValue" id="22" >
  <string name="MessageType" id="35" byteLength="1" value="X" />
  <UInt32 name="MsgSeqNum" id="34" byteLength="4" />
  <symbol name="Symbol" id="55" />
  <sequence name="MDEntries" >
    <length name="NoMDEntries" id="268" byteLength="1" />
    <string name="MDEntryType" id="269" byteLength="1" />
    <decimal name="MDEntryPx" id="270" byteLength="5" />
  </sequence>
</template>
```

10 – MD Entry Type

0	Bid
1	Ask
3	Index Value
5	Close
6	Settlement / EDSP Price

5.2 Heartbeat Message (Line Integrity Message) – Template ID 16

Heartbeat messages contain only a standard header. The heartbeat will repeat at a regular interval.

11 - Heartbeat Message Structure

Field ID	Field Name	Type	Length (Bytes)	Comments
	Standard Header			See table: <i>7 –Message Types</i> , MessageType = “0” (zero)

12 - Heartbeat Template

```
<template name="Heartbeat" id="16" >
  <string name="MessageType" id="35" byteLength="1" value="0" />
  <UInt32 name="MsgSeqNum" id="34" byteLength="4" />
</template>
```

5.3 Contributor Value Message – Template ID 23

Contributor Value messages contain values associated with a given symbol or index. They are transmitted when received by Cboe.

At least one MDEntry with MDEntryType == Index Value will always be present in the Contributor Value message. For indices where a Bid and Ask index value is calculated, MDEntryTypes Bid and Ask will be present. For indices where Bid and Ask is not calculated, MDEntryTypes for Bid and Ask are omitted.

13 – Contributor Value Message Structure

Field ID	Field Name	Type	Length (Bytes)	Comments
	Standard Header			See table: 7 –Message Types, MessageType = “X”
60	TransactTime	Unit64	8	Time that data was calculated or produced from client. Applies only to this message within the packet
55	Symbol	symbol		Index Symbol/iNAV Symbol
268	NoMDEntries	length	1	Number of MDEntries in this message. Will not exceed 255
Repeating Group - Fields Repeat NoMDEntries times				
269	MDEntryType	single byte string	1	Entry Type. See table: 10 – MD Entry Type
270	MDEntryPx	decimal	5	Index value associated with MDEntryType

14 – Contributor Value Template

```

<template name="ContributorValue" id="23"
  <string name="MessageType" id="35" byteLength="1" value="X"/>
  <uint32 name="MsgSeqNum" id="34" byteLength="4"/>
  <uint64 name="TransactTime" id="60" byteLength="8"/>
  <symbol name="Symbol" id="55"/>
  <sequence name="MDEntries">
    <length name="NoMDEntries" id="268" byteLength="1"/>
    <string name="MDEntryType" id="269" byteLength="1"/>
    <decimal name="MDEntryPx" id="270" byteLength="5"/>
  </sequence>
</template>

```

5.4 Index Summary – Template ID 24

Index Summary messages contain values associated with start or end of day values. An effective date is included for reference. Bid and ask fields are optional and not required.

At least one MDEntry with MDEntryType == Index Value will always be present in the Index Summary message. For indices where a Bid and Ask index value is calculated, MDEntryTypes Bid and Ask will be present. For indices where Bid and Ask is not calculated, MDEntryTypes for Bid and Ask are omitted.

15 – Index Summary Message Structure

Field ID	Field Name	Type	Length (Bytes)	Comments
	Standard Header			See table: 7 –Message Types, MessageType = “X”
60	TransactTime	Unit64	8	Time that data was calculated or produced from client. Applies only to this message within the packet

2400	EffectiveBusinessDate	string	8	YYYYMMDD format
20000	SummaryType	single byte string	1	type of summary message “1” SOD “2” EOD
55	Symbol	symbol		Index Symbol/iNAV Symbol
268	NoMDEntries	length	1	Number of MDEntries in this message. Will not exceed 255
Repeating Group - Fields Repeat NoMDEntries times				
269	MDEntryType	single byte string	1	Entry Type. See table: 10 – MD Entry Type
270	MDEntryPx	decimal	5	Index value associated with MDEntryType

16 – Index Summary Template

```

<template name="IndexSummary" id="24">
  <string name="MessageType" id="35" byteLength="1" value="X"/>
  <uint32 name="MsgSeqNum" id="34" byteLength="4"/>
  <uint64 name="TransactTime" id="60" byteLength="8"/>
  <string name="EffectiveBusinessDate" id="2400" byteLength="8"/>
  <length name="SummaryType" id="20000" byteLength="1"/>
  <symbol name="Symbol" id="55"/>
  <sequence name="MDEntries" >
    <length name="NoMDEntries" id="268" byteLength="1"/>
    <string name="MDEntryType" id="269" byteLength="1"/>
    <decimal name="MDEntryPx" id="270" byteLength="5"/>
  </sequence>
</template>

```

5.5 Index Value With Status – Template ID 25

Index Value With Status messages contain the values associated with a calculated index. They are transmitted when an index is calculated.

Market data associated with the index are contained in MDEntries with an MDEntryType indicating the type of market data. The Index Value With Status message may include any of the following MDEntryTypes (See table 10 – MD Entry Type).

If MDEntryType == ‘Index Value’, the message may also have a Bid/Ask value but will not have ‘Settlement/EDSP’ or ‘Close’ prices. If a message contains either a ‘Settlement/EDSP’ or ‘Close’ MDEntryType, the message will not contain any other MDEntrytypes.

17 – Index Value with Status Message Structure

Field ID	Field Name	Type	Length (Bytes)	Comments
	Standard Header			See table: <i>7 –Message Types</i> , MessageType = “X”
20001	Index Status	single byte string	1	N = Normal I = Indicative
55	Symbol	symbol		Index Symbol / iNAV Symbol
268	NoMDEntries	length	1	Number of MDEntries in this message. Will not exceed 255
Repeating Group - Fields Repeat NoMDEntries times				
269	MDEntryType	single byte string	1	Entry Type. See table: 10 – MD Entry Type 0 = Bid 1 = Ask 3 = Index Value 5 = Close Value 6 = Settlement / EDSP Price
270	MDEntryPx	decimal	5	Index value associated with MDEntryType

18 – Index Value with Status Template

```

<template name="IndexValueWithStatus" id="25" >
  <string name="MessageType" id="35" byteLength="1" value="X" />
  <uint32 name="MsgSeqNum" id="34" byteLength="4" />
  <length name="IndexStatus" id="20001" byteLength="1" />
  <symbol name="Symbol" id="55" />
  <sequence name="MDEntries" >
    <length name="NoMDEntries" id="268" byteLength="1" />
    <string name="MDEntryType" id="269" byteLength="1" />
    <decimal name="MDEntryPx" id="270" byteLength="5" />
  </sequence>
</template>

```

5.6 Index Definition Message - Template ID 26

An Index Definition message will be published for every index within each cycle, repeating on a regular interval no longer than 15 minutes. Messages within each cycle will be published at a rate of 4 per second (subject to change). Any index status changes occurring during an in-progress cycle will be reflected in the next cycle.

Clients are required to capture at Current Record Number = 1 to receive the complete series of Index Definition messages.

Field ID	Field Name	Type	Length (Bytes)	Comments
	Standard Header			See table: <i>7 –Message Types</i> , MessageType = “D”
	Current Record Number	uint32	4	Clients should begin capturing data beginning at 1.

Cboe Global Indices Feed Specification (Version 1.4.9)

911	Total Number of Records	uint32	4	Total number of updates within loop.
	Symbol	symbol		External index symbol.
	Description	string	Variable	Description of the index.
	Date	string	10	YYYY-MM-DD format.
	Status	unit32	1	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	two byte string	2	Originating source of the index: <blank> = Not Specified CO = Cboe CC = Cboe Custom TP = Third-party
Repeating Group				
	Channel	string	Variable	Sequence Group of the following types: Cboe Global Indices Feed Main = Main CGI = Strategy, Benchmark, and Custom Indices MSCI = Morgan Stanley INAV = Net Asset Values FTSE = Russell Indices MSTAR = Morningstar MSTARC = Morningstar Custom CCCY = Cryptocurrency

6 Appendix A – Multicast Group and Port Information

Please contact the [Cboe NOC](#) to be able access the Cboe Global Indices Feed platform certification, production, and DR environments.

Data Channel	Location	Type	RP	Source IP	IP	Port
Cboe Global Indices Feed Main	NY5 Prod DC	Primary	74.115.128.168	174.136.169.32/29	224.0.131.168	30201
		Secondary	74.115.128.169	174.136.169.40/29	233.130.124.168	30201
	DR DC	Primary	170.137.16.130	170.137.16.192/29	233.182.199.8	31201
	NY5 Cert DC	Primary	74.115.128.166	174.136.160.32/29	224.0.74.176	32201
MSCI	NY5 Prod DC	Primary	74.115.128.168	174.136.169.32/29	224.0.131.169	30201
		Secondary	74.115.128.169	174.136.169.40/29	233.130.124.169	30201
	DR DC	Primary	170.137.16.130	170.137.16.192/29	233.182.199.9	31201
	NY5 Cert DC	Primary	74.115.128.166	174.136.160.32/29	224.0.74.177	32201
FTSE	NY5 Prod DC	Primary	74.115.128.168	174.136.169.32/29	224.0.131.170	30201
		Secondary	74.115.128.169	174.136.169.40/29	233.130.124.170	30201
	DR DC	Primary	170.137.16.130	170.137.16.192/29	233.182.199.10	31201
	NY5 Cert DC	Primary	74.115.128.166	174.136.160.32/29	224.0.74.178	32201
CCCY	NY5 Prod DC	Primary	74.115.128.168	174.136.169.32/29	224.0.131.171	30201
		Secondary	74.115.128.169	174.136.169.40/29	233.130.124.171	30201
	DR DC	Primary	170.137.16.130	170.137.16.192/29	233.182.199.11	31201
	NY5 Cert DC	Primary	74.115.128.166	174.136.160.32/29	224.0.74.179	32201
INAV	NY5 Prod DC	Primary	74.115.128.168	174.136.169.32/29	224.0.131.172	30201
		Secondary	74.115.128.169	174.136.169.40/29	233.130.124.172	30201
	DR DC	Primary	170.137.16.130	170.137.16.192/29	233.182.199.12	31201
	NY5 Cert DC	Primary	74.115.128.166	174.136.160.32/29	224.0.74.180	32201
MSTAR	NY5 Prod DC	Primary	74.115.128.168	174.136.169.32/29	224.0.131.173	30201
		Secondary	74.115.128.169	174.136.169.40/29	233.130.124.173	30201
	DR DC	Primary	170.137.16.130	170.137.16.192/29	233.182.199.13	31201
	NY5 Cert DC	Primary	74.115.128.166	174.136.160.32/29	224.0.74.181	32201

Cboe Global Indices Feed Specification (Version 1.4.9)

MSTARC	NY5 Prod DC	Primary	74.115.128.168	174.136.169.32/29	224.0.131.174	30201
		Secondary	74.115.128.169	174.136.169.40/29	233.130.124.174	30201
	DR DC	Primary	170.137.16.130	170.137.16.192/29	233.182.199.14	31201
	NY5 Cert DC	Primary	74.115.128.166	174.136.160.32/29	224.0.74.182	32201
CGI	NY5 Prod DC	Primary	74.115.128.168	174.136.169.32/29	224.0.131.175	30201
		Secondary	74.115.128.169	174.136.169.40/29	233.130.124.175	30201
	DR DC	Primary	170.137.16.130	170.137.16.192/29	233.182.199.15	31201
	NY5 Cert DC	Primary	74.115.128.166	174.136.160.32/29	224.0.74.183	32201
Cboe Global Indices Definition Feed	NY5 Prod DC	Primary	74.115.128.168	174.136.169.32/29	224.4.7.0	30201
		Secondary	74.115.128.169	174.136.169.40/29	234.170.137.0	30201
	DR DC	Primary	170.137.16.130	170.137.16.192/29	224.4.7.128	31201
	NY5 Cert DC	Primary	74.115.128.166	174.136.160.32/29	234.170.137.128	32201

7 Appendix B – Examples

7.1 Understanding the Hex Data Diagrams

Throughout this section are hexadecimal printouts of data from a channel. Each example is in a similar format.

Offset	Hexadecimal data 4 bytes per group	ASCII representation
00000000	CD580000 00030000 0134C8E5 A8992B6C	.X.....4....+1
00000010	031FFE00 00006600 00006401 00f...d..

The first column of numbers is the zero-based byte offset of the first hex byte on that line, expressed as a hexadecimal offset. The next 4 columns of data are hexadecimal values for the bytes, with 4 bytes per group. Spaces are for formatting purposes only. Characters between pipes (“|”) are ascii representations of the hex data. Periods indicate non-printable values.

The first 16 bytes of each hex dump is the packet header.

7.2 Packet Header Example

1 - Packet Header Hex Dump

00000000 01038B00 000135A6 B387070B 00000000 5.....
--

2 - Packet Header Decoded

SendTime(02/22 14:14:37.831)

BlkVersion(1) BlkSize(907) MsgsInBlock(11) FirstMsgSeq(0)

7.3 Index Example

3 - Index Hex Dump

00000000 01002F00 00015023 AC595A01 00000002 .../...P#.YZ.....
00000010 001F1658 00000002 034F4558 0333FE00 ...X.....OEX.3..
00000020 014B6030 FE00014B 4E31FE00 014B71 ..K`0...KN1...Kq

4 - Index Decoded

Message#(1) MsgSize(31)			
TemplateId(22) TemplateName(IndexValue)	AppMsg:	IndexValue	Type: X
FieldType FieldName		Id	Value
sbSTRING: MessageType		35	X
UINT32: MsgSeqNum		34	2
STRING: Symbol		55	OEX
SEQUENCE: NoMDEntries		268	Len=3
MSG_GROUP NumElems: 2			
sbSTRING: MDEntryType		269	3
uDECIMAL: MDEntryPx		270	848.32
MSG_GROUP NumElems: 2			
sbSTRING: MDEntryType		269	0

Cboe Global Indices Feed Specification (Version 1.4.9)

uDECIMAL:	MDEntryPx	270	848.14
MSG_GROUP NumElems:	2		
sbSTRING:	MDEntryType	269	1

7.4 Heartbeat Message

3 - Heartbeat Hex Dump

00000000	01001800	000135A7	00C6C901	00000F955.....
00000010	00081030	00000F95			...0....

4 - Heartbeat Decoded

Message#(1) MsgSize(8)			
TemplateId(16)	AppMsg: Heartbeat Type: 0		
FieldType	FieldName	Id	Value
sbSTRING:	MessageType	35	0
UINT32:	MsgSeqNum	34	3989

Support

Questions regarding this document can be directed to the Cboe Operations Support Center (“OSC”) at 312.786.7635 or indexsupport@cboe.com. The latest version of this document can be found on the [Indices microsite](#).

Revision History

Date	Version	Description of Change
09/17/15	1.0	New document
12/07/15	1.1	Updated the cover page to reference indices
12/10/15	1.2	Updated MD-EntryType table reference.
06/07/16	1.2	Updated the Index Options market hours start time from 5:00 a.m. to 2:00 a.m.
08/30/17	1.3	Updated Appendix A – Multicast Group and Port Information. Include FTSE, Cryptocurrency, and INAV.
09/01/17	1.3.1	Updated primary and secondary multicast group IPs for new Cboe Global Indices Feed platform.
10/17/17	1.3.2	Cboe branding/logo changes.
02/06/18	1.3.3	Updated date for parallel testing over 5 channels.
02/28/22	1.3.4	Added notes to support Contributor API protocol (effective 05/01/22).
03/11/22	1.3.5	Updated comments for <i>Symbol/field</i> (55) to “Index Symbol/iNAV Symbol”.
06/01/22	1.4	CSMI rebranded to Cboe Global Indices Feed. All references are updated (effective 06/02/22). MSTAR and MSTARC channels added to Multicast Group and Port Information.
06/14/22	1.4.1	Updated MSTARC Secondary IP Address.
06/30/22	1.4.2	Minor updates to “Index Value” message and “Contributor Value” message references.
07/26/22	1.4.3	Minor formatting updates.
08/08/22	1.4.4	Updated hours of operation.
02/28/23	1.4.5	Added new CGI Index Channel and Index Values with Status messages.
03/13/23	1.4.6	Added new Cboe Global Indices Definition Feed Channel and new Index Definition message (effective 03/27/23).
03/16/23	1.4.7	Updated Index Definition “MessageType=D”. Updated Index Definition publication rate to 4 per second. Updated section 5.5 to indicate that if MDEntryType == ‘Index Value’, the message may also have a Bid/Ask value but will not have ‘Settlement/EDSP’ or ‘Close’ prices.
04/14/23	1.4.8	Updated CGI NY5 cert IP address. Updated CGI dissemination end time to 10:30 a.m. CT.
08/16/23	1.4.9	Updated Symbol field character length. Updated Character Array String information. In Channel field of <i>Index Definition</i> message, updated CSM to “Cboe Global Indices Feed Main=Main” and updated CGI to “CGI=Strategy, Benchmark, and Custom Indices”.