Cboe Japan Multicast Depth of Book (PITCH) Specification

Version 1.0.3

August 11, 2023

Contents

1 Introduction

1.1 Overview

This specification is the standard Multicast Depth of Book (PITCH) specification for the Cboe Japan platforms. Note that this specification will be the standard Multicast PITCH specification to be used for both the Cboe Japan Alpha PTS and the Cboe Japan Select PTS.

Participants may use the Multicast PITCH protocol to receive real-time trading information directly from Cboe Japan. The Multicast PITCH protocol provides symbol information, real-time depth of book quotations, and execution information direct from Cboe Japan. Participants can connect to the Multicast PITCH feed from 5 a.m. to 6 p.m. JST.

Cboe Japan PITCH cannot be used to enter orders. For order entry, refer to the Cboe Japan FIX or BOE Specification.

All versions of the Multicast PITCH feed will be Gig-shaped and will be available from one or both Cboe Japan datacenters. Participants may choose to take one or more of the following Multicast PITCH feeds depending on their location and connectivity to Cboe Japan.

Multicast PITCH Feed Descriptions:

Market	Shaping	Served From Data Center (Primary/Secondary)	Multicast Feed ID
Alpha	Gig	Primary	JAM – Feed A
Alpha	Gig	Primary	JBM – Feed B
Alpha	Gig	Secondary	JEM – Feed E
Select	Gig	Primary	SAM – Feed A
Select	Gig	Primary	SBM – Feed B
Select	Gig	Secondary	SEM – Feed E

1.2 Feed Connectivity

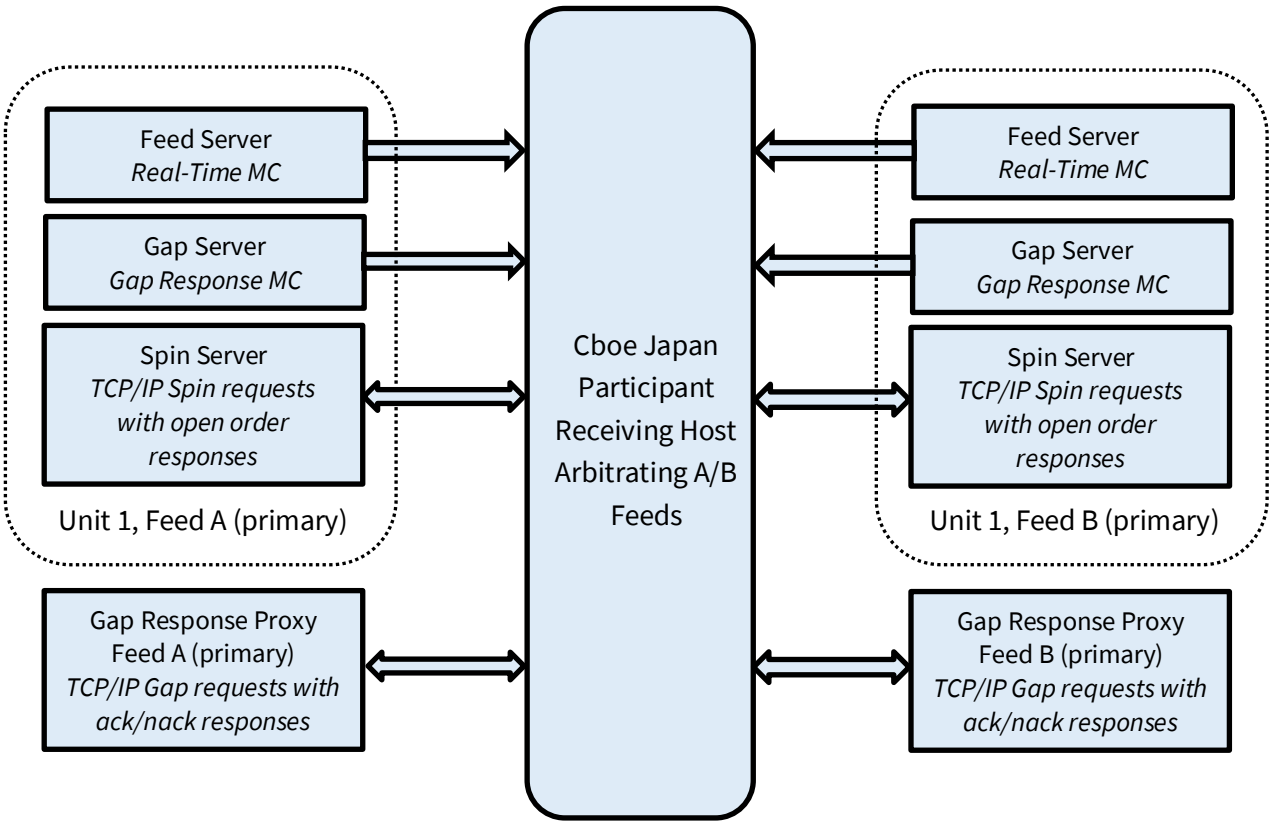
PITCH feeds are available to participants who connect to Cboe Japan via cross-connect, dedicated circuit, or a supported carrier.

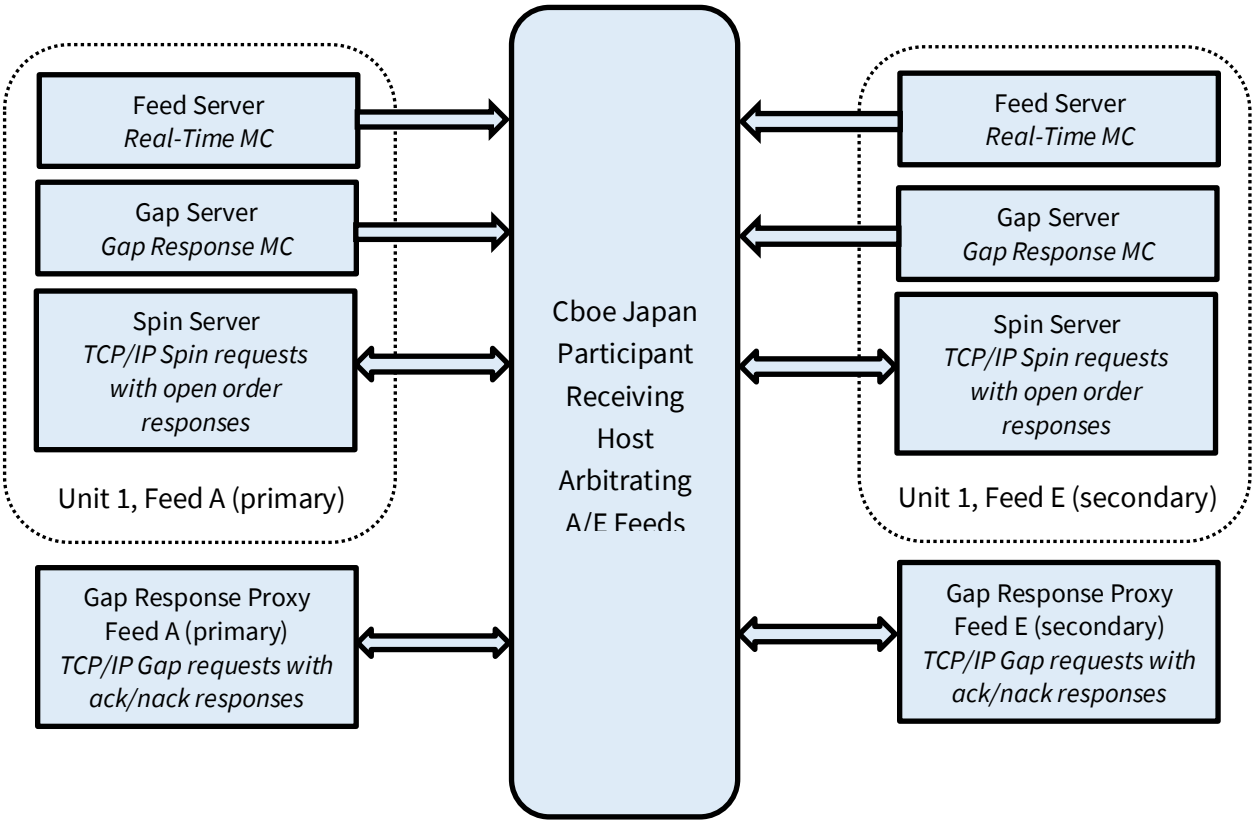
Participants with sufficient connectivity may choose to take both the A and B feeds from Cboe Japan's primary datacenter and arbitrate the feeds to recover lost data. Alternatively, participants may choose to arbitrate feeds from both datacenters. It should be noted that feeds from the secondary datacenter will have additional latency compared to those connected with Cboe Japan in the primary datacenter due to proximity and business continuity processing.

When arbitrating, participants can utilize the fact the redundant feeds have messages that are sequenced and process the next expected sequence from whichever feed it's received from first. The A and B feeds are created utilizing distinct infrastructure, and the architecture is such that neither the A nor B feed should be expected to advantaged relative to the other (i.e., performance should be generally equal). Any duplicate message sequence can be dropped. Arbitration reduces the chances of losing a message due to packet loss.

Multicast PITCH real-time events are delivered using a published range of multicast addresses divided into units, each with a unique range of symbols. A TCP/IP connection to one of Cboe Japan's Gap Request Proxy ("GRP") servers can be used to request dropped messages. Replayed messages are delivered on a separate set of multicast ranges reserved for packet retransmission. Intraday, a spin of all open orders may be requested from a Spin Server. This allows a participant to become current without requesting a gap for all messages up to that point in the day.

The following diagram is a logical representation Multicast PITCH feed message flow between Cboe Japan and a participant feed handler listening to the "A", "B", and "E" instances of a unit:





1.3 Symbol Ranges, Units, and Sequence Numbers

Symbols will be separated into units, and the symbol distribution will not change intra-day. Cboe Japan does, however, reserve the right to add multicast addresses or change the symbol distribution. Participants will be notified and provided sufficient time to conform with the changes. Care should be taken to ensure that address changes, address additions, and symbol 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 *or more* units will be delivered on a single multicast address. As with symbol ranges, unit distribution across multicast addresses will not change intra-day but may change after notice has been given.

Symbol distribution across units as well as unit distribution across multicast addresses are identical for real-time and gap response multicast addresses.

1.4 Gap Request Proxy and Message Retransmission

Requesting delivery of missed sequenced data is achieved by establishing a TCP connection to a Cboe Japan GRP port and then receiving requested messages on designated gap recovery multicast addresses. Participants who do not wish to request missed messages do not need to connect to a GRP port for any reason or listen to the multicast addresses reserved for message retransmission. Participants choosing to request missed data will need to connect to their assigned GRP port, log in, and request gap ranges as necessary. All gap requests will be responded to with a `Gap Response` message. A `Gap Response` *Status* code of "A" (accepted) signals that the replayed messages will be delivered via the appropriate gap response multicast address. Any other `Gap Response` *Status* code will indicate the reason that the request cannot be serviced.

The GRP limits gap requests by message count, frequency, and age. Gap requests will only be serviced if they are within a defined sequence range of the current multicast sequence number for the requested unit. Participants will receive a total daily allowance of gap requested messages. In addition, each participant is given renewable one-second and one-minute gap request limits. If the gap allowances are exceeded, the gap request will be rejected as defined here4.5. The participant can then wait until the time-based gap request limits reset or perform a spin as defined here1.5. If the daily allowance of gap requests is exceeded the participant must perform a spin.

If overlapping gap requests are received within a short period of time all requests will receive a successful `Gap Response` message from the GRP, but the gap server will send the union of the sequence ranges across grouped gap requests. Participants will receive gap responses for their requested unit/sequence/count, but receivers should be prepared for the gap responses to be delivered via multicast in non-contiguous blocks.

Gap acknowledgements or rejects will be delivered to participants for every gap request received by the GRP. Participants should be prepared to see replayed multicast data before or after the receipt of the gap response acknowledgement from the GRP.

Here shows an example flow of messages between a participant and Cboe Japan's Multicast PITCH feed, Gap Server, and Gap Request Proxy.

1.5 Spin Servers

A Spin Server is available for each unit. The server allows participants to connect via TCP and receive a spin of the current order book and symbols on that unit. By using the spin, a participant can get the current Cboe Japan book quickly in the middle of the trading session without worry of gap request limits. The Spin Server for each unit is assigned its own address and/or TCP port.

Upon successful login and periodically thereafter, a `Spin Image Available` message is sent which contains a sequence number indicating the most recent message applied to the book. Using a `Spin Request` message, a participant may request a spin for the orders up to a sequence number noted within one of the *last ten* `Spin Image Available` messages distributed. If the `Spin Request` submitted does not present a sequence number that matches one of the last ten `Spin Image Available` messages distributed, the spin will return orders up to the next closest sequence number reported through a `Spin Image Available` message that is greater than the sequence number requested.

In the case a participant sends a sequence number in a `Spin Request` that is higher than the sequence number reported by the most recent `Spin Image Available` message, the next spin image to be generated will be returned when it is available. If the requested sequence number is still higher at that time, an "O" (Out of Range) error will be generated.

A spin consists only of `Trading Status` and `Add Order` messages. `Trading Status` messages will be sent in spins for all symbols that are not closed ("C"), which results in at least one message for every symbol that has not been closed ("C") since system start-up. Spins will not contain any message for an order which is no longer on the book. While receiving the spin, the participant must buffer multicast messages received. If the `Spin Image Available` message sequence number is the participant's reference point, multicast messages with larger sequence numbers should be buffered. If a non-`Spin Image Available` sequence number is the participant's reference point which they send in their `Spin Request`, they should buffer from that point on. However, the participant should then disregard all messages from the feed server that are not greater than the sequence number in the `Spin Response`. When a `Spin Finished` message is received, the buffered messages must be applied to the spun copy of the book to bring it current.

Here shows an example flow of messages between a participant and Cboe Japan Multicast PITCH feed and Spin Server.

Participants are required to send Heartbeat messages to Cboe Japan no less than every 5 seconds, even while a spin response is in progress. Failure to do so is the most common cause of participant difficulties while processing spin responses, especially during periods of high market activity. Cboe Japan recommends that participants send a heartbeat every second to stay well within this heartbeat requirement.

2 Protocol

Cboe Japan Participants may use the PITCH protocol over multicast to receive real-time full depth of book quotations and execution information direct from Cboe Japan.

2.1 Message Format

The messages that make up the PITCH protocol are delivered using the `Sequenced Unit Header` which handles sequencing and delivery integrity. All messages delivered via multicast as well as to/from the Gap Request Proxy ("GRP") will use the `Sequenced Unit Header` for handling message integrity.

All UDP delivered events will be self-contained. Developers can assume that UDP delivered data will not cross frame boundaries and a single Ethernet frame will contain only one `Sequenced Unit Header` with associated data.

TCP/IP delivered events from the GRP may cross frames as the data will be delivered as a stream of data with the TCP/IP stack controlling Ethernet framing.

The PITCH feed is comprised of a series of dynamic length sequenced messages. Each message begins with *Length* and *Message Type* fields. Cboe Japan 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.

2.2 Data Types

The following field types are used within the `Sequenced Unit Header`, GRP messages, Spin Server messages, and PITCH.

- **Alphanumeric** fields are left justified ASCII fields and space padded on the right.
- **Binary** fields are unsigned and sized to "Length" bytes and ordered using Little Endian convention (least significant byte first).
- **Binary Price** fields are unsigned Little Endian encoded 8 byte binary fields with 4 implied decimal places (denominator = 10,000).
- **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).
- **Printable ASCII** fields are left justified ASCII fields that are space padded on the right that may include ASCII values in the range of 0x20 – 0x7e.
- **Binary UTC Timestamp** are 8 byte unsigned Little Endian values representing the number of nanoseconds since the epoch (00:00:00 UTC on 1 January 1970).

2.3 Message Framing

PITCH messages will be combined into a single UDP frame where possible to decrease message overhead and total bandwidth. The count of messages in a UDP frame will be communicated using the `Sequenced Unit Header`. Framing will be determined by the server for each unit and site. The content of the multicast across feeds (e.g., A/B) will be identical, **but framing will not be consistent across feeds**. Receiving processes that receive and arbitrate multiple feeds cannot use frame level arbitration to fill gaps.

2.4 Sequenced Unit Header

The `Sequenced Unit Header` is used for all Multicast PITCH messages as well as messages to and from the Gap Request Proxy ("GRP") and Spin Servers.

Both sequenced and unsequenced data may be delivered using the `Sequenced Unit Header`. Unsequenced headers will have a 0 value for the *Hdr Sequence* field and potentially for the *Hdr Unit* field. All messages sent to and from the GRP and Spin Servers are unsequenced while multicast may contain both sequenced and unsequenced messages.

Sequenced messages have implied sequences with the first message having the sequence number contained in the header. Each subsequent message will have an implied sequence one greater than the previous message up to a maximum of count messages. Multiple messages can follow a `Sequenced Unit Header`, but a combination of sequenced and unsequenced messages cannot be sent within one header.

The sequence number for the first message in the next frame can be calculated by adding the *Hdr Count* field to the *Hdr Sequence*. This technique will work for sequenced messages and `Heartbeats`.

Sequenced Unit Header				
Field	**Offset**	**Length**	**Value/Type**	**Description**
Hdr Length	0	2	Binary	Length of entire block of messages. Includes this header and Hdr Count messages to follow.
Hdr Count	2	1	Binary	Number of messages to follow this header.
Hdr Unit	3	1	Binary	Unit that applies to messages included in this header.
Hdr Sequence	4	4	Binary	Sequence of first message to follow this header.
Total Length = 8 bytes				

2.5 Heartbeat Messages

The `Sequenced Unit Header` with a *Hdr Count* field set to "0" will be used for heartbeat messages. During trading hours heartbeats will be sent from the GRP, Spin Server, and all multicast addresses if no data has been delivered within one second. Heartbeat messages never increment the sequence

number for a unit but can be used to detect gaps on the real-time multicast channels during low update rate periods.

Heartbeats on the real-time multicast addresses during trading hours will have a *Hdr Sequence* value equal to the sequence of the next sequenced message to be sent for the unit. Heartbeats on gap multicast addresses will always have the *Hdr Sequence* field set to 0. All heartbeat messages sent to and from the GRP and Spin Server are unsequenced and should have *Hdr Sequence* and *Hdr Unit* fields set to 0.

Outside of trading hours Cboe Japan sends heartbeats on all real-time and gap channels with a sequence of 0 to help participants validate multicast connectivity. Heartbeats might not be sent outside of normal trading hours during scheduled maintenance.

Cboe Japan expects heartbeats to be sent to the GRP and Spin Servers on live connections no less than every 5 seconds. Failure to receive two consecutive heartbeats will result in the GRP or Spin Server terminating the participant connection. This also applies when the participant is receiving a spin from the Spin Server, the heartbeats must continue to be sent from the participant to the Spin Server.

2.6 Execution IDs and Order IDs

Execution IDs and Order IDs that are reported in PITCH may be converted to base 36 and then matched to Execution IDs and Order IDs that are received over FIX or BOE acknowledgements. Conversion rules and examples are provided to allow for participants to match these ID types.

2.6.1 Execution IDs

Convert to nine-character, base 36, zero-padded on the left. Binary values represented in Little Endian format.

Binary Value (Hex)	Decimal (base 10)	Cboe Base36 Value
24 45 20 30 15 00 00 00	91001734436	015T02ZOK
8B 0F FF 6E 27 00 00 00	169365933963	025T03ROR

2.6.2 Order IDs

Convert to 12-character, base 36. No padding should be required. Binary values represented in Little Endian format.

Binary Value (Hex)	Decimal (base 10)	Cboe Base36 Value
00 60 A3 58 6C 5E 29 40	288958144494319104	27174309PSLC
09 AC 22 D4 83 8A EF 22	157336438470486729	17174206VA2X

3 PITCH Messages

The PITCH messages reflect the state of Cboe Japan including the order addition, order deletion, order modification, or execution of an order in the system.

3.1 Unit Clear

The `Unit Clear` message instructs feed recipients to clear all orders for the Cboe Japan book in the unit specified in the `Sequenced Unit Header`. It would be distributed in rare recovery events such as a datacenter fail-over.

Unit Clear				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x97	`Unit Clear` Message
Reserved	2	4	Binary	Reserved (undefined)
Total Length = 6 bytes				

3.2 Trading Status

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 Cboe Japan system:

- C = Closed. Not accepting orders. Implied at system start-up for all symbols.
- T = Trading. Continuous trading session open. Accepting orders.
- H = Halted. Not accepting orders. Existing orders may be cancelled.
- h = Halted in Pre-open. Indicates that the security is halted in pre-market hours. If the halt continues into the continuous trading session, the trading status will transition from “h” to “H”.

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

Trading Status				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	Length of this message including this field
Message Type	1	1	0x50	`Trading Status` message
Timestamp	2	8	Binary UTC Timestamp	Nanoseconds since epoch
Symbol	10	6	Alphanumeric	Symbol (right padded with spaces).
Trading Status	16	1	Alphanumeric	`C` = Closed `T` = Trading `H` = Halted `h` = Halted in Pre-Open
Trading Status Flags	17	1	Bit Field	**Bit 0 – Short sell price check** `0` = Short sell price check not in effect `1` = Short sell price check in effect
Total Length = 18 bytes				

3.3 Add Order Message

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

Add Order				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x51	`Add Order` Message
Timestamp	2	8	Binary UTC Timestamp	Nanoseconds since epoch
Order Id	10	8	Binary	Day-specific identifier assigned to this order. Order Ids received on PITCH may be compared to those received on order acknowledgements in FIX or BOE by converting the decimal (base 10) value to a base 36 value. Example conversion: Base 10 – 1079067412513217551 Base 36 – 874XH1UZEHOV
Side Indicator	18	1	Alphanumeric	`B` = Buy Order `S` = Sell Order
Quantity	19	4	Binary	Number of shares being added to the book.
Symbol	23	6	Alphanumeric	Symbol (right padded with spaces).

Price	29	8	Binary Price	The display price of the order.
Reserved	37	1	Binary	Reserved (undefined)
Total Length = 38 bytes				

3.4 Order Modification Messages

Order Modification messages refer to an *Order Id* previously sent with an `Add Order` message. Multiple Order Modification messages may modify a single order and the effects are cumulative. Modify messages may update the size and/or the price of an order on the book. When the remaining size of an order reaches zero, the order is dead and should be removed from the book.

3.4.1 Order Executed Message

`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*.

Order Executed				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x52	`Order Executed` Message
Timestamp	2	8	Binary UTC Timestamp	Nanoseconds since epoch
Order Id	10	8	Binary	*Order Id* of a previously sent `Add Order` message that was executed.
Executed Quantity	18	4	Binary	Number of shares executed.
Execution Id	22	8	Binary	Cboe Japan generated day-unique execution identifier of this execution.
Contra Order Id	30	8	Binary	*Order Id* of the contra order to this execution. The contra order id may not have been previously sent in an Add `Order` message.
Tick Direction	38	1	Printable ASCII	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 ("+").
Total Length = 39 bytes				

3.4.2 Reduce Size Message

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

Reduce Size				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field
Message Type	1	1	0x39	`Reduce Size` Message
Timestamp	2	8	Binary UTC Timestamp	Nanoseconds since epoch
Order Id	10	8	Binary	*Order Id* of a previously sent `Add Order` message that has been reduced.
Cancelled Quantity	18	4	Binary	Number of shares cancelled.
Total Length = 22 bytes				

3.4.3 Modify Order Message

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.

Modify				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field
Message Type	1	1	0x3A	`Modify Order` Message
Timestamp	2	8	Binary UTC Timestamp	Nanoseconds since epoch
Order Id	10	8	Binary	*Order Id* of a previously sent `Add Order` message that has been modified.
Quantity	18	4	Binary	Number of shares associated with this order after this modify (may be less than the number entered).
Price	22	8	Binary Price	The order price after this modify.
Reserved	30	1	Binary	Reserved (undefined)
Total Length = 31 bytes				

3.4.4 Delete Order Message

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.

Delete				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x3C	`Delete Order` Message
Timestamp	2	8	Binary UTC Timestamp	Nanoseconds since epoch
Order Id	10	8	Binary	*Order Id* of a previously sent `Add Order` message that has been removed from order book.
Total Length = 18 bytes				

3.5 Trade Message

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

A `Trade` message is sent when there is an execution against any non-displayed portion of an iceberg order. As with visible orders, iceberg orders may be executed in parts. A complete view of all Cboe Japan executions can be built by combining all `Order Executed` and `Trade` messages.

Trade				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field
Message Type	1	1	0x53	`Trade` Message
Timestamp	2	8	Binary UTC Timestamp	Nanoseconds since epoch
Symbol	10	6	Alphanumeric	Symbol (right padded with spaces).
Quantity	16	4	Binary	Incremental number of shares executed or reported.
Price	20	8	Binary Price	The price of the trade.
Execution Id	28	8	Binary	Cboe Japan generated day-unique execution identifier of this execution. *Execution Id* is also referenced in the `Trade Break` message.
Order Id	36	8	Binary	*Order Id* of the executed order.
Contra Order Id	44	8	Binary	*Order Id* of the contra order that executed with this order.
Total Length = 52 bytes				

3.6 Trade Break Message

The `Trade Break` message is sent whenever an execution on Cboe Japan is cancelled. Applications that simply build a Cboe Japan book can ignore `Trade Break` messages.

Trade Break				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field
Message Type	1	1	0x3E	`Trade Break` Message
Timestamp	2	8	Binary UTC Timestamp	Nanoseconds since epoch
Execution Id	10	8	Binary	Cboe Japan generated day-unique identifier of the execution that was broken. *Execution Id* refers to previously sent `Order Executed` or `Trade` message.
Total Length = 18 bytes				

3.7 End of Session

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

End of Session				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field
Message Type	1	1	0x2D	`End of Session` Message
Reserved	2	4	Binary	Reserved (undefined)
Total Length = 6 bytes				

4 Gap Request Proxy Messages

The following messages are used for initializing a TCP/IP connection to the Gap Request Proxy ("GRP") and to request message retransmissions. Participants only need to implement the following messages if gap requests will be made. Each of the following message types must be wrapped by an unsequenced `Sequenced Unit Header` as described here2.4. The following messages will not be delivered using multicast.

Participants are advised to login to the GRP service at start of day in readiness to request the recovery of gaps as they occur. Please note that the recoverable window of messages advances throughout the day.

4.1 Login

The `Login` message is the first message sent to the GRP by the participant process after the connection to the GRP is established. Failure to login before sending any other message type will result in the connection being dropped by the GRP.

Login				
Field	**Offset**	**Length**	**Value/Type**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x01	`Login` Message
SessionSubId	2	4	Alphanumeric	*SessionSubId* supplied by Cboe Japan.
Username	6	4	Alphanumeric	*Username* supplied by Cboe Japan.
Filler	10	2	Alphanumeric	(space filled)
Password	12	10	Alphanumeric	*Password* supplied by Cboe Japan.
Total Length = 22 bytes				

4.2 Login Response

The `Login Response` message is sent by the GRP to the participant process in response to a `Login` message. The *Status* field is used to reflect an accepted login or the reason the session was not accepted. If login fails, the connection will be dropped after the `Login Response` message is sent.

Login Response				
Field	**Offset**	**Length**	**Value/Type**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x02	`Login Response` Message
Status	2	1	Alphanumeric	Accepted or reason for reject.
Total Length = 3 bytes				

Login Response – Status Codes	
'A'	Login Accepted
'N'	Not authorized (Invalid Username/Password)
'B'	Session in use
'S'	Invalid Session

4.3 Heartbeat

Heartbeat messages must be sent once every five seconds in order to keep the participant's connection to the GRP server alive. Heartbeat messages are sent using the `Sequenced Unit Header` as described here.

4.4 Gap Request

The `Gap Request` message is used by a participant's process to request retransmission of a sequenced message (or messages) by one of Cboe Japan's gap servers.

Gap Request				
Field	**Offset**	**Length**	**Value/Type**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x03	`Gap Request` Message
Unit	2	1	Binary	*Unit* that the gap is requested for.
Sequence	3	4	Binary	*Sequence* of first message (Lowest sequence in range).
Count	7	2	Binary	*Count* of messages requested.
Total Length = 9 bytes				

4.5 Gap Response

The `Gap Response` message is sent by the GRP in response to a `Gap Request` message. The *Unit* and *Sequence* fields will match the values supplied in the `Gap Request` message. A `Gap Response` message, with a *Status* of Accepted or reason for failure, will be sent for each `Gap Request` message received by the GRP.

Gap Response				
Field	**Offset**	**Length**	**Value/Type**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x04	`Gap Response` Message.
Unit	2	1	Binary	*Unit* the gap was requested for.
Sequence	3	4	Binary	*Sequence* of first message in request.
Count	7	2	Binary	*Count* of messages requested.
Status	9	1	Alphanumeric	Accepted or reason for reject*.
Total Length = 10 bytes				

Gap Response – Status Codes	
'A'	Accepted
'O'	Out of range (ahead of sequence or too far behind)
'D'	Daily gap request allocation exhausted
'M'	Minute gap request allocation exhausted
'S'	Second gap request allocation exhausted
'C'	Count request limit for one gap request exceeded
'I'	Invalid Unit specified in request
'U'	Unit is currently unavailable

* - All non-'A' status codes should be interpreted as a reject.

4.6 Gap Server Usage Example

The following diagram shows the exchange of messages over time between a participant and Cboe Japan's Multicast PITCH feed, Gap Request Proxy, and Gap Server.

At time 0 assume the participant state of the book is current through sequence 310170, and the next expected sequence is 310171.

At time 1 the participant sends a `Login` message to the Gap Request Proxy (GRP) server and at time 2 receives a `Login Response` message indicating the login has been accepted. The participant is now successfully logged into the GRP and able to request gaps. Note this is just for example purposes and in practice the participant is encouraged to log into the GRP at the start of the trading day.

At time 3 and 4, the participant receives sequences 310171 and 310172. These messages are in sequence and the participant applies these messages to their book. The state of the book is current through sequence 310172 and the next expected sequence is 310173.

At time 5 and 6, the participant receives sequences 310176 and 310177 and determines sequences 310173 through 310175 are missing (i.e., a gap was detected). Sequences 310176 and 310177 are then cached for later use.

At time 7 a `Gap Request` message is sent to the GRP to request the missing messages, starting at sequence 310173 for a total of 3 messages.

At time 8, the participant receives sequence 310178. Since there are still missing sequences, it cannot apply this message to the book and sequence 310178 is cached for later use.

At time 9, the participant receives a `Gap Response` message from the GRP indicating the gap request was successful and it can expect the requested messages to be sent from the Gap Server.

At time 10, the participant receives sequence 310179. Since there are still missing sequences, it cannot apply this message to the book and sequence 310179 is cached for later use.

At time 11, the participant receives sequence 310173 from the Gap Server. Since the last sequence applied was 310172, the participant can apply this message to the book. The state of the book is current through sequence 310173 and the next expected sequence is 310174.

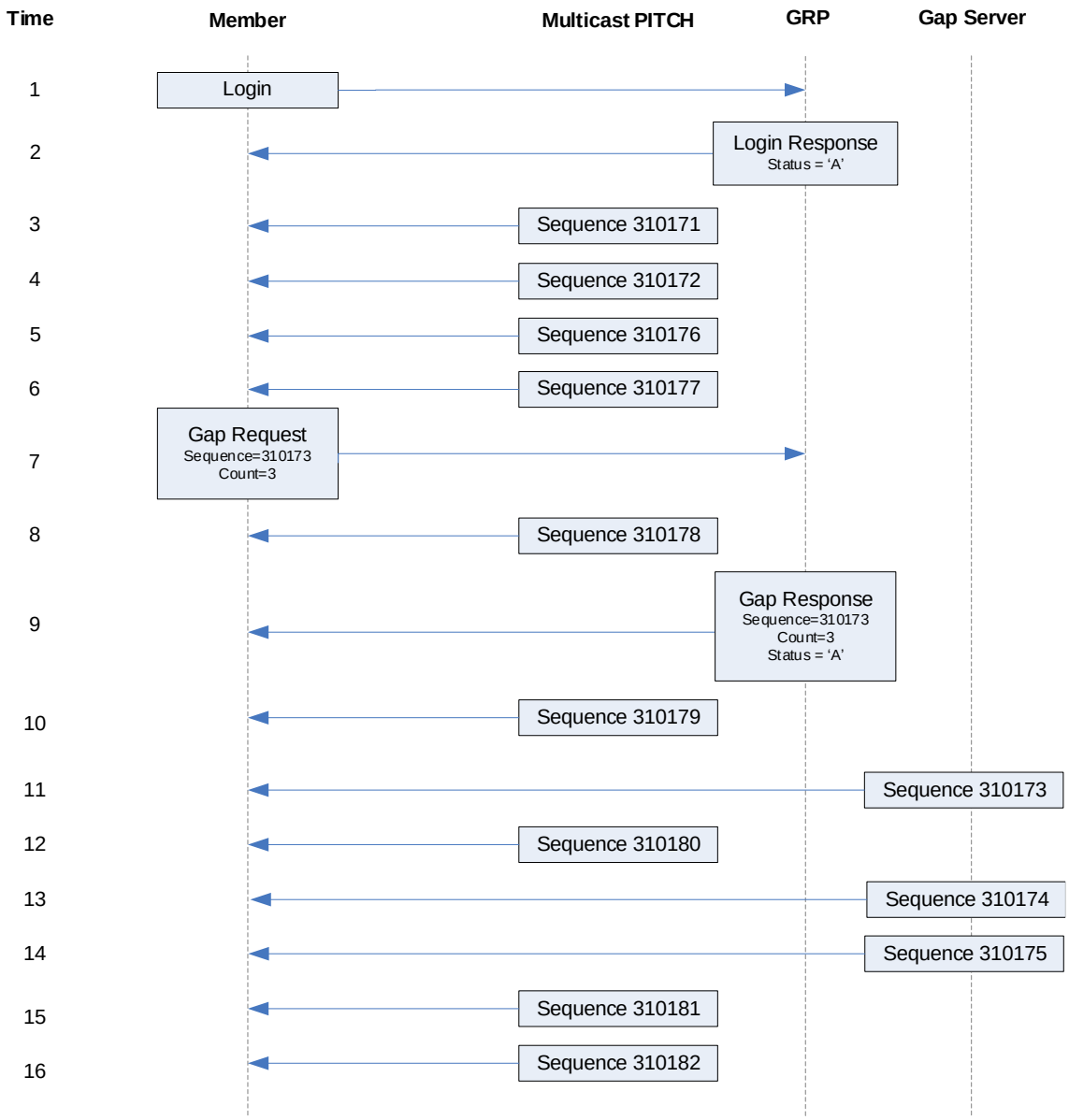
At time 12, the participant receives sequence 310180. Since there are still missing sequences, it cannot apply this message to the book and sequence 310180 is cached for later use.

At time 13 and 14, the participant receives sequences 310174 and 310175 from the Gap Server. Since the last sequence applied was 310173, the participant can apply these messages to the book.

Now that all the missing sequences have been received from the Gap Server, the participant can apply the cached sequence messages 310176 through 310180. At this point the book should be current with the PITCH feed. The state of the book is current through sequence 310180 and the next expected sequence is 310181.

At times 15 and 16, sequences 310181 and 310182 are received. Since there are no missing sequences, and these messages are in sequence, the participant applies these messages to their book. The state of the book is current through sequence 310182 and the next expected sequence is 310183.

It should be noted that other participants may also request gaps, and the participants should be prepared to ignore any message from the Gap Server it is not expecting or does not need.



5 Spin Messages

5.1 Login

The `Login` message is the first message sent to the Spin Server by a participant's process after the connection to the Spin Server is established. Failure to login before sending any other message type will result in the connection being dropped by the Spin Server.

The format of the `Login` message for the Spin Server is identical to that of the GRP as described previously here.

5.2 Login Response

The `Login Response` message is sent by the Spin Server to a participant's process in response to a `Login` message. The status field is used to reflect an accepted login or the reason the session was not accepted. If login fails, the connection will be dropped after the `Login Response` message is sent.

The format of the `Login Response` message for the Spin Server is identical to that of the GRP described previously here.

5.3 Heartbeat

Heartbeat messages must be sent once every five seconds in order to keep the participant's connection to the spin server alive. Heartbeat messages are sent using the `Sequenced Unit Header` as described here and here2.5.

5.4 Spin Image Available

The `Spin Image Available` message is sent once per second and indicates through what sequence number a spin is available.

Spin Image Available				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x80	`Spin Image Available` Message
Sequence	2	4	Binary	Spin is available which is current through this sequence number.
Total Length = 6 bytes				

5.5 Spin Request

The `Spin Request` message is used by a participant's process to request transmission of a spin of the unit's order book. Refer here for more complete details regarding *Sequence* specification and buffering requirements.

Spin Request				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x81	`Spin Request` Message
Sequence	2	4	Binary	Sequence number from a `Spin Image Available` message received by the participant.
Total Length = 6 bytes				

5.6 Spin Response

The `Spin Response` message is sent in response to a participant's `Spin Request` message indicating whether a spin will be sent.

Spin Response				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x82	`Spin Response` Message
Sequence	2	4	Binary	Sequence number from a `Spin Image Available` message.
Order Count	6	4	Binary	Number of `Add Order` messages which will be contained in this spin.
Status	10	1	Alphanumeric	Accepted or reason for reject*.
Total Length = 11 bytes				

Spin Response - Status Codes	
'A'	Accepted
'O'	Out of Range (*Sequence* requested is greater than *Sequence* available by the next spin)
'S'	Spin already in progress (only one spin can be running at a time)

* - All non-'A' status codes should be interpreted as a reject.

5.7 Spin Finished

The `Spin Finished` message is sent to indicate that all messages for the spin requested have been sent. A `Spin Finished` message is only sent if a `Spin Request` was not rejected. Upon receipt of a `Spin Finished` message, any buffered multicast messages should be applied to the participant's copy of the book to make it current.

Spin Finished				
Field Name	**Offset**	**Length**	**Type/(Value)**	**Description**
Length	0	1	Binary	*Length* of this message including this field.
Message Type	1	1	0x83	`Spin Finished` Message

Sequence	2	4	Binary	Sequence number from the `Spin Response` message.
Total Length = 6 bytes				

5.8 Spin Server Usage Example

The following diagram (see next page) shows the exchange of messages over time between a participant and Cboe Japan's Multicast PITCH feed and Spin Server. The spin will consist of `Trading Status` and `Add Order` messages.

At time 1, the participant has no state of the book and desires to become current. The participant caches the received Multicast PITCH messages (sequences 310172 and 310173) for later use. Since the participant has no book, they cannot yet be applied.

At time 5, the participant has successfully logged into the Spin Server and has cached another message, sequence 310174.

At time 7, the participant receives a `Spin Image Available` message which indicates that the spin server can give them a spin of all open orders as of sequence 310169. The participant does not have all messages cached after 310169 (they are missing 310170 and 310171), so this spin is not useful to the participant.

At time 10, the participant receives a `Spin Image Available` message which is useful since it would be a spin of all orders up to and including sequence 310175 and the participant has all messages after 310175 cached.

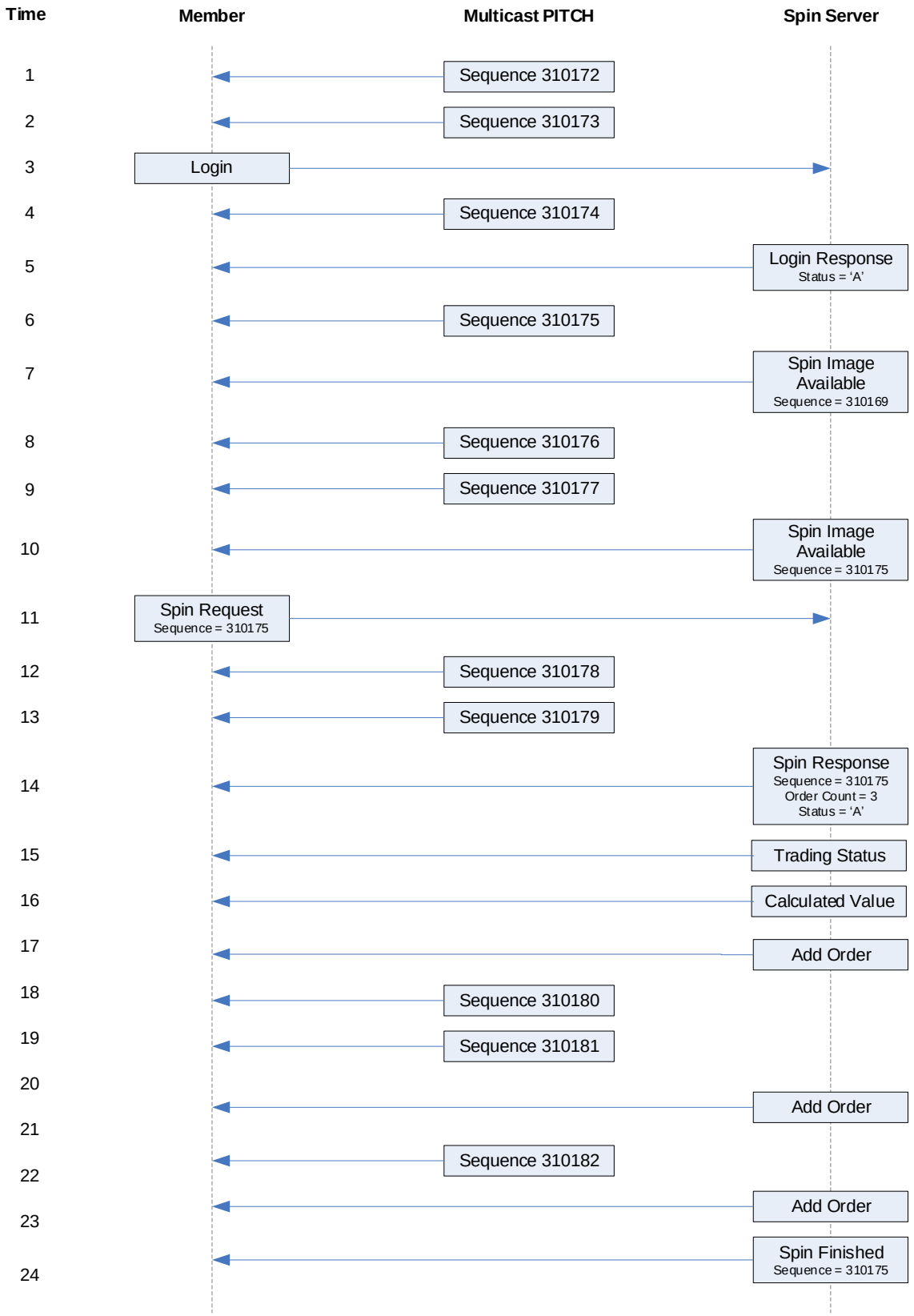
At time 11, the participant sends a `Spin Request` for all messages up to and including 310175 and continues to cache Multicast PITCH messages received.

At time 14, the spin server acknowledges the spin request and indicates that three open orders will be sent.

At time 24, the spin server indicates that it has finished sending all open orders. The participant must then apply the cached messages from sequence number 310176 through current.

Notes:

- Spin Servers are available for each unit. Participants may need to employ multiple Spin Servers depending upon their architecture.



6 Message Types

6.1 Gap Request Proxy Messages

0x01	Login
0x02	Login Response
0x03	Gap Request
0x04	Gap Response

6.2 Spin Server Messages

0x01	Login
0x02	Login Response
0x80	Spin Image Available
0x81	Spin Request
0x82	Spin Response
0x83	Spin Finished

6.3 PITCH Messages

0x97	Unit Clear
0x50	Trading Status
0x51	Add Order
0x52	Order Executed
0x39	Reduce Size
0x3A	Modify Order
0x3C	Delete Order
0x53	Trade
0x3E	Trade Break
0x2D	End of Session

7 Example Messages

7.1 Individual Messages

Each of the following message types must be wrapped by a sequenced or unsequenced `Sequenced Unit Header` as described here2.4. Note that in the following examples, each byte is represented by two hexadecimal digits.

7.1.1 Login Message

Length	16	22 bytes
Type	01	Login
SessionSubId	30 30 30 31	"0001"
Username	46 49 52 4D	"FIRM"
Filler	20 20	" "
Password	41 42 43 44 30 30 20 20 20 20	"ABCD00 "

7.1.2 Login Response Message

Length	03	3 bytes
Type	02	Login Response
Status	41	Login accepted

7.1.3 Gap Request Message

Length	09	9 bytes
Type	03	Gap Request
Unit	01	Unit 1
Sequence	3B 10 00 00	First message: 4155
Count	32 00	50 messages

7.1.4 Gap Response Message

Length	10	10 bytes
Type	04	Gap Response
Unit	01	Unit 1
Sequence	3B 10 00 00	First message: 4155
Count	32 00	50 messages
Status	41	Accepted

7.1.5 Unit Clear

Length	06	6 bytes
Type	97	Unit Clear
Reserved	20 20 20 20	(Reserved)

7.1.6 Trading Status

Length	12	18 bytes
Type	50	Trading Status
Timestamp	F0 77 BB CE 2A 6A 62 16	1612968348641622000 ns since epoch (Feb 10 2021 14:45:48 GMT+0)
Symbol	34 36 38 39 20 20	“4689 “
Trading Status	54	T = Trading
Short Sell	01	1 = Short sell price check is active.

7.1.7 Add Order

Length	26	38 bytes
Type	51	Add Order
Timestamp	F0 77 BB CE 2A 6A 62 16	1612968348641622000 ns since epoch
Order Id	05 40 5B 77 8F 56 1D 0B	631WC4000005 (base36)
Side Indicator	42	B = Buy
Quantity	BC 02 00 00	700 shares
Symbol	34 36 38 39 20 20	“4689 “
Price	A8 5E BC 00 00 00 00 00	1234.5000
Reserved	00	(Reserved)

7.1.8 Order Executed

Length	27	39 bytes
Type	52	Order Executed
Timestamp	F0 77 BB CE 2A 6A 62 16	1612968348641622000 ns since epoch
Order Id	05 40 5B 77 8F 56 1D 0B	631WC4000005 (base36)
Executed Quantity	BC 02 00 00	700 shares
Execution Id	34 2B 46 E0 BB 00 00 00	0AAP09VEC (base36)
Contra Order Id	72 41 5B 77 8F 56 1D 0B	631WC40000AA (base36)
Tick Direction	2B	“+” tick higher

7.1.9 Reduce Size

Length	16	22 bytes
Type	39	Reduce Size
Timestamp	F0 77 BB CE 2A 6A 62 16	1612968348641622000 ns since epoch
Order Id	05 40 5B 77 8F 56 1D 0B	631WC4000005 (base36)
Cancelled Quantity	BC 02 00 00	700 shares

7.1.10 Modify Order

Length	1F	31 bytes
Type	3A	Modify Order
Timestamp	F0 77 BB CE 2A 6A 62 16	1612968348641622000 ns since epoch
Order Id	05 40 5B 77 8F 56 1D 0B	631WC4000005 (base36)
Quantity	BC 02 00 00	700 shares
Price	A8 5E BC 00 00 00 00 00	1234.5000
Reserved	00	(Reserved)

7.1.11 Delete Order

Length	12	18 bytes
Type	3C	Delete Order
Timestamp	F0 77 BB CE 2A 6A 62 16	1612968348641622000 ns since epoch
Order Id	05 40 5B 77 8F 56 1D 0B	631WC4000005 (base36)

7.1.12 Trade

Length	34	52 bytes
Type	53	Trade
Timestamp	F0 77 BB CE 2A 6A 62 16	1612968348641622000 ns since epoch
Symbol	34 36 38 39 20 20	"4689 "
Quantity	BC 02 00 00	700 shares
Price	A8 5E BC 00 00 00 00 00	1234.5000
Execution Id	34 2B 46 E0 BB 00 00 00	0AAP09VEC (base36)
Order Id	05 40 5B 77 8F 56 1D 0B	631WC4000005 (base36)
Contra Order Id	09 40 5B 77 8F 56 1D 0B	631WC4000009 (base36)

7.1.13 Trade Break

Length	12	18 bytes
Type	3E	Trade Break
Timestamp	F0 77 BB CE 2A 6A 62 16	1612968348641622000 ns since epoch
Execution Id	34 2B 46 E0 BB 00 00 00	0AAP09VEC (base36)

7.1.14 End of Session

Length	06	6 bytes
Type	2D	End of Session
Reserved	00 00 00 00	(Reserved)

7.2 Order Entry Examples

The following examples demonstrate the Cboe Japan Multicast PITCH messages sent in response to various orders entered in Cboe Japan.

7.2.1 Reduce Size Example

Action	Message Description
A visible order to buy 1000 shares of "2345" at 10.00 is entered. The order is assigned Id 100000000001 and rests on the book. An Add Order message with price of 10.00 and quantity of 100 is sent.	Type: 51 (Add Order) Timestamp: 1612968348641622000 (nanos) Order Id (base36): 100000000001 Side: B (Buy) Quantity: 1000 Symbol: "2345 " Price: 10.00 Reserved: 0 (Reserved)
The size of the order is modified by the participant from 1000 to 900. A Modify Order is sent with the new price of 11.00.	Type: 39 (Reduce Size) Timestamp: 1612968348641623000 (nanos) Order Id (base36): 100000000001 Quantity: 100 Reserved: 0 (Reserved)

7.2.2 Modify Order Example

Action	Message Description
A visible order to buy 1000 shares of "2345" at 10.00 is entered. The order is assigned Id 100000000001 and rests on the book. An Add Order message with price of 10.00 and quantity of 100 is sent.	Type: 51 (Add Order) Timestamp: 1612968348641622000 (nanos) Order Id (base36): 100000000001 Side: B (Buy) Quantity: 1000 Symbol: "2345 " Price: 10.00 Reserved: 0 (Reserved)
The price of the order is modified by the participant from 10.00 to 11.00. A Modify Order is sent with the new price of 11.00.	Type: 3A (Modify Order) Timestamp: 1612968348641623000 (nanos) Order Id (base36): 100000000001 Quantity: 1000 Price: 11.00 Reserved: 0 (Reserved)

7.2.3 Iceberg Order Execution Example

Action	Message Description
An iceberg order to buy 1000 shares of "2345" at 10.00 is entered, with a display quantity of 500. The order is assigned Id 100000000005 and rests on the book. An Add Order message with price of 10.00 and quantity of 500.	Type: 51 (Add Order) Timestamp: 1612968348641622000 (nanos) Order Id (base36): 100000000005 Side: B (Buy) Quantity: 500 Symbol: "2345 " Price: 10.00 Reserved: 0 (Reserved)
An order to sell 200 shares of "2345" at 10.00 is entered. This order is executed against the resting iceberg buy order. An Order Executed message is sent with price of 10.00 and quantity of 200. The remaining visible quantity of the iceberg order is 300, with the total quantity of 800.	Type: 52 (Order Executed) Timestamp: 1612968348641623000 (nanos) Order Id (base36): 100000000005 Executed Quantity: 200 Execution Id (base36): 01000000C Contra Order Id (base36): 100000000006 Tick Direction: + (price ticked higher)
Another visible order to sell shares of "2345" at 10.00 is entered. This order is executed against the resting iceberg buy order. An Order Executed message is sent for the	Type: 52 (Order Executed) Timestamp: 1612968348641624000 (nanos) Order Id (base36): 100000000005 Executed Quantity: 300 Execution Id (base36): 01000000D

remaining visible quantity of 300. At this point the visible quantity of the order is exhausted and participants would remove the order from their books.	Contra Order Id (base36): 100000000007 Tick Direction: U (no change since an uptick)
A Trade message is sent for the hidden quantity of the iceberg order.	Type: 53 (Trade) Timestamp: 1612968348641624000 (nanos) Symbol: "2345 " Quantity: 500 Price: 10.00 Execution Id (base36): 01000000E Order Id: 100000000008 (obfuscated) Contra Order Id (base36): 100000000007

7.2.4 Iceberg Order Replenished Example

Action	Message Description
An iceberg order to buy 750 shares of "2345" at 10.00 is entered, with a display quantity of 500. The order is assigned Id 100000000009 and rests on the book. An Add Order message with price of 10.00 and quantity of 500.	Type: 51 (Add Order) Timestamp: 1612968348641622000 (nanos) Order Id (base36): 100000000009 Side: B (Buy) Quantity: 500 Symbol: "2345 " Price: 10.00 Reserved: 0 (Reserved)
A visible order to sell 500 shares of "2345" at 10.00 is entered. This order is executed against the resting iceberg buy order. An Order Executed message is sent with price of 10.00 and quantity of 500. At this point the visible quantity of the order is exhausted and participants would remove the order from their books.	Type: 52 (Order Executed) Timestamp: 1612968348641623000 (nanos) Order Id (base36): 100000000009 Executed Quantity: 500 Execution Id (base36): 01000000F Contra Order Id (base36): 10000000000A Tick Direction: "-" (price ticked lower)
The iceberg order is replenished with the remaining quantity of 250 shares. An Add Order is sent with an obfuscated (new) order Id.	Type: 51 (Add Order) Timestamp: 1612968348641623000 (nanos) Order Id (base36): 10000000000B (obfuscated) Side: B (Buy) Quantity: 250 Symbol: "2345 " Price: 10.00 Reserved: 0 (Reserved)

8 Multicast Configuration

8.1 Production Environment Configuration

8.1.1 Limitations/Configurations

The following table defines the configuration for network and gap request limitations. These limitations are session-based. Cboe Japan reserves the right to adjust the gap request limitations to improve the effectiveness of the gap request infrastructure.

Period/Type	Limit/Setting	Notes
MTU	1500	Cboe Japan will send UDP messages up to 1500 bytes. Participants should ensure their infrastructure is configured accordingly.
Gap Response Delay	2 ms	The Gap Server will delay resending sequenced messages via multicast for the specified limit to satisfy multiple GRP gap requests with one multicast response.
Count	100	Any single gap request may not be for more than this number of dropped messages.
1 Second	320 Requests	Maximum number of retransmission requests allowed per second for each session, renewed every clock second.
1 Minute	1,500 Requests	Maximum number of retransmission requests allowed per minute for each session, renewed every clock minute.
Day	100,000 Requests	Maximum number of retransmission requests allowed per day for each session.
Within Range	1,000,000 Messages	Participants' retransmission requests must be within this many messages of the most recent sequence sent by the real-time feed per session.

8.1.2 Unit/Symbol Distribution

The following table describes the Cboe Japan symbol distribution across units.

Unit	Symbol Range
1	0000 – 4999
2	5000 – 9999

Note – Cboe Japan reserves the right to add multicast addresses with prior notice, but no migration period. Notice will be given that the distribution will change on a certain date. Care should be taken to support mappings in these tables via software configuration.

8.1.3 Cboe Japan Multicast Routing Parameters

Data Center	Rendezvous Point
Alpha Primary Data Center A feed	74.115.128.15
Alpha Primary Data Center B feed	74.115.128.16
Select Primary Data Center A feed	74.115.128.17
Select Primary Data Center B feed	74.115.128.18
Alpha DR Data Center E Feed	74.115.128.19
Select DR Data Center E Feed	74.115.128.20

For additional information about physical connectivity, refer to the Cboe Japan Connectivity Manual.

8.1.4 Alpha(+Match) Address/Unit Distribution

The following tables describe the unit distribution across the Cboe Japan PITCH feeds for the Alpha(+Match) market.

Primary Data Center		Gig-Shaped "A" Feed [JAM] 170.137.202.0/28		Gig-Shaped "B" Feed [JBM] 170.137.202.16/28	
Unit	IP Port	Real-time MC	Gap Resp. MC	Real-time MC	Gap Resp. MC
1	30501	233.218.133.124	233.218.133.125	233.218.133.126	233.218.133.127
2	30502				

Note – Cboe Japan reserves the right to add multicast addresses with prior notice, but no migration period. Notice will be given that the distribution will change on a certain date. Care should be taken to support mappings in these tables via software configuration.

Secondary Data Center		Gig-Shaped "E" Feed [JEM] 170.137.205.0/28	
Unit	IP Port	Real-time MC	Gap Resp. MC
1	31501	233.218.133.132	233.218.133.133
2	31502		

Note – Cboe Japan reserves the right to add multicast addresses with prior notice, but no migration period. Notice will be given that the distribution will change on a certain date. Care should be taken to support mappings in these tables via software configuration.

8.1.5 Select Address/Unit Distribution

The following tables describe the unit distribution across the Cboe Japan PITCH feeds for the Select market.

Primary Data Center		Gig-Shaped "A" Feed [SAM] 170.137.203.0/28		Gig-Shaped "B" Feed [SBM] 170.137.203.16/28	
Unit	**IP Port**	**Real-time MC**	**Gap Resp. MC**	**Real-time MC**	**Gap Resp. MC**
1	30501	233.218.133.128	233.218.133.129	233.218.133.130	233.218.133.131
2	30502				

Note – Cboe Japan reserves the right to add multicast addresses with prior notice, but no migration period. Notice will be given that the distribution will change on a certain date. Care should be taken to support mappings in these tables via software configuration.

Secondary Data Center		Gig-Shaped "E" Feed [SEM] 170.137.205.128/28	
Unit	**IP Port**	**Real-time MC**	**Gap Resp. MC**
1	31501	233.218.133.134	233.218.133.135
2	31502		

Note – Cboe Japan reserves the right to add multicast addresses with prior notice, but no migration period. Notice will be given that the distribution will change on a certain date. Care should be taken to support mappings in these tables via software configuration.

8.2 Certification Environment Configuration

8.2.1 Unit/Symbol Distribution

The following table describes the Cboe Japan symbol distribution across units.

Unit	Symbol Range
1	0000 – 4999
2	5000 – 9999

Note – Cboe Japan reserves the right to add units and/or change symbol distribution with 48 hours of notice and no migration period. Notice will be given that the distribution will change on a certain date. Care should be taken to support mappings in these tables via software configuration.

8.2.2 Certification Multicast Routing Parameters

Primary Certification Data Center	Rendezvous Point
Primary Data Center feed	74.115.128.14

8.2.3 Alpha(+Match) Address/Unit Distribution

The following tables describe the unit distribution across the certification Cboe Japan PITCH feeds for the Alpha(+Match) market.

Primary Data Center		CertFeed [Cert] 170.137.204.128/28	
Unit	**IP Port**	**Real-time MC**	**Gap Resp. MC**
1	32501	233.218.133.120	233.218.133.121
2	32502		

Note – Cboe Japan reserves the right to add multicast addresses with prior notice, but no migration period. Notice will be given that the distribution will change on a certain date. Care should be taken to support mappings in these tables via software configuration.

8.2.4 Select Address/Unit Distribution

The following tables describe the unit distribution across the certification Cboe Japan PITCH feeds for the Select market.

Primary Data Center		CertFeed [Cert] 170.137.204.144/28	
Unit	**IP Port**	**Real-time MC**	**Gap Resp. MC**
1	32501	233.218.133.122	233.218.133.123
2	32502		

Note – Cboe Japan reserves the right to add multicast addresses with prior notice, but no migration period. Notice will be given that the distribution will change on a certain date. Care should be taken to support mappings in these tables via software configuration.

9 Connectivity

9.1 Supported Extranet Carriers

Cboe Japan may certify a number of carriers to redistribute Multicast data feeds, as defined in the Cboe Japan Connectivity Manual. For more information on receiving Cboe Japan Multicast PITCH through any of these providers, please contact the vendor noted in the Extranet Providers section of the Connectivity Manual.

9.2 Bandwidth Recommendation

The Gig-shaped feeds require 1 Gbps of bandwidth. Cboe Japan will use 90% of these respective bandwidths for Multicast PITCH to allow participants to use the same physical connection for order entry if desired.

10 Support

Please direct questions or comments regarding this specification to TradeDeskJP@cboe.com.

Revision History

Document Version	Date	Description
1.0.0	22/12/13	Initial version.
1.0.1	23/02/10	Updated “Halted in Pre-open” state value to “h”. Updated message type for `Trade` message to 0x52. Corrected example `Trade` timestamp values in section 7.2.3. Corrected example `Add Order` timestamp values in section 7.2.4.
1.0.2	23/03/03	Updated multicast IP addresses throughout Section 8.
1.0.3	23/08/11	Updated short sell indicator in Trading Status message to properly describe it as a bit field with a single bit in use. Updated Cboe Japan Multicast Routing Parameters and Certification Multicast Routing Parameters.