



Cboe Australia Multicast Depth of Book (PITCH) Specification

Version 1.0.6

7 November 2022

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.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 5 |
| 1.1 | Overview | 5 |
| 1.2 | Feed Connectivity | 6 |
| 1.3 | Symbol Ranges, Units, and Sequence Numbers | 8 |
| 1.4 | Gap Request Proxy and Message Retransmission | 8 |
| 1.5 | Spin Servers..... | 9 |
| 2 | Protocol | 11 |
| 2.1 | Message Format | 11 |
| 2.2 | Data Types | 11 |
| 2.3 | Message Framing | 12 |
| 2.4 | Sequenced Unit Header | 12 |
| 2.5 | Heartbeat Messages..... | 12 |
| 2.6 | Execution IDs and Order IDs..... | 13 |
| 2.6.1 | Execution IDs..... | 13 |
| 2.6.2 | Order IDs..... | 13 |
| 3 | PITCH Messages | 14 |
| 3.1 | Unit Clear | 14 |
| 3.2 | Trading Status | 14 |
| 3.3 | Add Order Message | 15 |
| 3.4 | Order Modification Messages..... | 16 |
| 3.4.1 | Order Executed Message..... | 16 |
| 3.4.2 | Reduce Size Message..... | 17 |
| 3.4.3 | Modify Order Message | 17 |
| 3.4.4 | Delete Order Message | 18 |
| 3.5 | Trade Message..... | 18 |
| 3.6 | Trade Break Message..... | 20 |
| 3.7 | Calculated Value Message..... | 20 |
| 3.8 | End of Session | 21 |
| 4 | Gap Request Proxy Messages..... | 22 |
| 4.1 | Login | 22 |
| 4.2 | Login Response | 22 |
| 4.3 | Heartbeat..... | 23 |
| 4.4 | Gap Request..... | 23 |
| 4.5 | Gap Response..... | 23 |
| 4.6 | Gap Server Usage Example | 24 |
| 5 | Spin Messages..... | 27 |
| 5.1 | Login..... | 27 |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | | |
|----------|--|-----------|
| 5.2 | Login Response | 27 |
| 5.3 | Heartbeat | 27 |
| 5.4 | Spin Image Available | 27 |
| 5.5 | Spin Request..... | 27 |
| 5.6 | Spin Response | 28 |
| 5.7 | Spin Finished | 28 |
| 5.8 | Spin Server Usage Example | 29 |
| 6 | Message Types | 31 |
| 6.1 | Gap Request Proxy Messages | 31 |
| 6.2 | Spin Server Messages | 31 |
| 6.3 | PITCH Messages | 31 |
| 7 | Example Messages..... | 32 |
| 7.1 | Individual Messages..... | 32 |
| 7.1.1 | Login Message | 32 |
| 7.1.2 | Login Response Message | 32 |
| 7.1.3 | Gap Request Message..... | 32 |
| 7.1.4 | Gap Response Message | 32 |
| 7.1.5 | Unit Clear..... | 32 |
| 7.1.6 | Trading Status..... | 33 |
| 7.1.7 | Add Order..... | 33 |
| 7.1.8 | Order Executed | 33 |
| 7.1.9 | Reduce Size | 33 |
| 7.1.10 | Modify Order..... | 34 |
| 7.1.11 | Delete Order | 34 |
| 7.1.12 | Trade (On-Exchange Electronic Execution) | 34 |
| 7.1.13 | Trade (Off-Exchange Trade Report)..... | 34 |
| 7.1.14 | Trade Break | 35 |
| 7.1.15 | Calculated Value | 35 |
| 7.1.16 | End of Session | 35 |
| 7.2 | Order Entry Examples..... | 36 |
| 7.2.1 | Modify Order Example..... | 36 |
| 7.2.2 | Undisclosed Order Execution Example | 36 |
| 7.2.3 | Iceberg Order Execution Example | 37 |
| 7.2.4 | Iceberg Order Replenished Example | 37 |
| 8 | Multicast Configuration | 39 |
| 8.1 | Production Environment Configuration..... | 39 |
| 8.1.1 | Limitations/Configurations | 39 |
| 8.1.2 | Unit/Symbol Distribution..... | 40 |
| 8.1.3 | CXA Multicast Routing Parameters | 40 |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | | |
|-----------|--|-----------|
| 8.1.4 | CXA Address/Unit Distribution | 40 |
| 8.2 | Certification Environment Configuration..... | 41 |
| 8.2.1 | Unit/Symbol Distribution..... | 41 |
| 8.2.2 | Certification Multicast Routing Parameters | 41 |
| 8.2.3 | CXA Address/Unit Distribution | 41 |
| 9 | Connectivity | 42 |
| 9.1 | Supported Extranet Carriers..... | 42 |
| 9.2 | Bandwidth Recommendation | 42 |
| 10 | Support..... | 43 |

1 Introduction

1.1 Overview

This specification is the standard Multicast Depth of Book (PITCH) specification for the Cboe Australia (“CXA”) platform.

Clients may use the Multicast PITCH protocol to receive real-time trading information directly from CXA. The Multicast PITCH protocol provides symbol information, real-time depth of book quotations, and execution information direct from CXA.

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

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

Multicast PITCH Feed Descriptions:

| Shaping | Served From Data Centre (Primary/Secondary) | Multicast Feed ID |
|---------|--|----------------------|
| Gig | Primary | AAM – Feed A |
| Gig | Primary | ABM – Feed B |
| Gig | Secondary | AEM – Feed E |

1.2 Feed Connectivity

PITCH feeds are available to clients who connect to CXA via cross-connect, dedicated circuit, or a supported carrier.

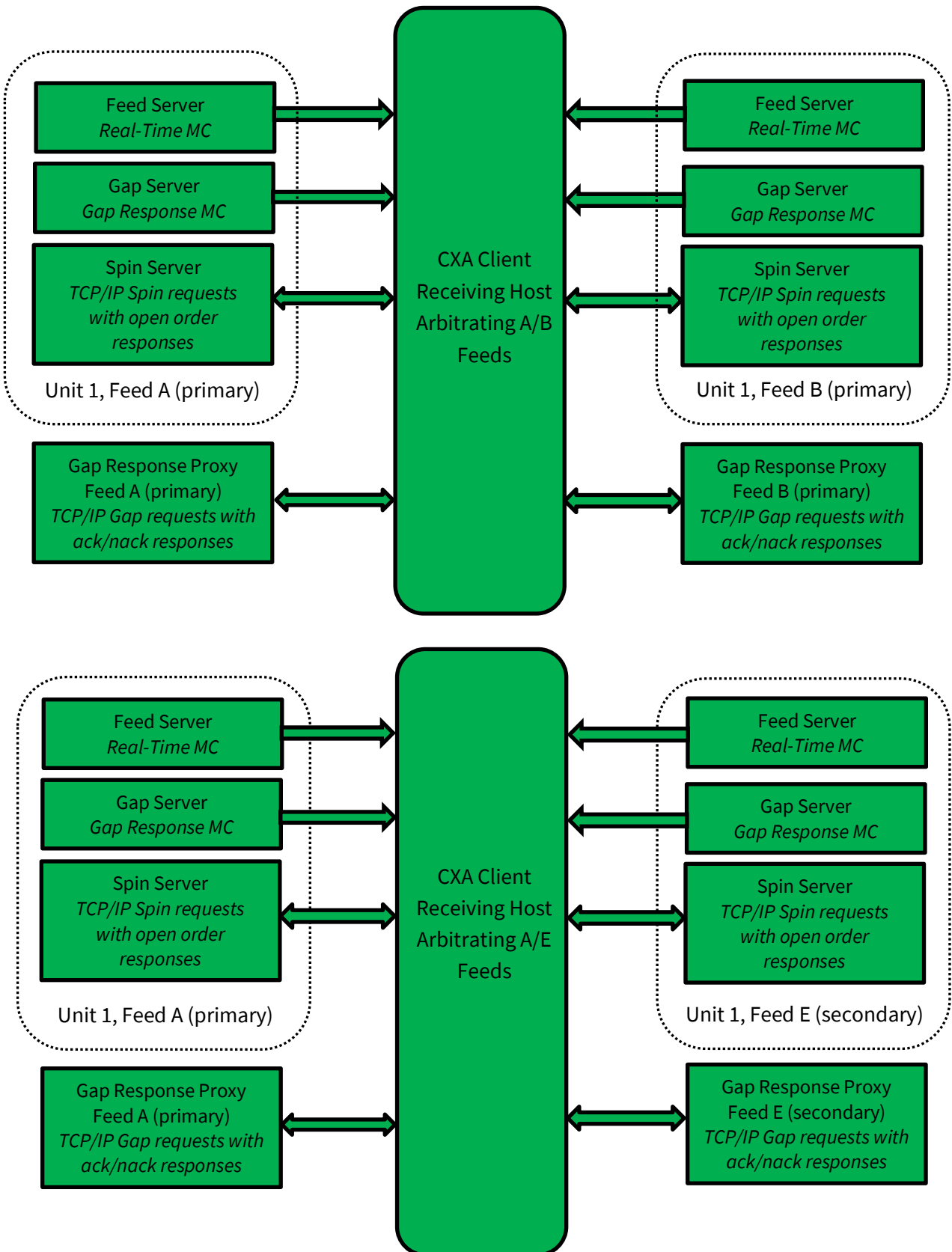
Clients with sufficient connectivity may choose to take both the A and B feeds from CXA's primary datacentre and arbitrate the feeds to recover lost data. Alternatively, clients may choose to arbitrate feeds from both datacentres. It should be noted that feeds from the secondary datacentre will have additional latency compared to those connected with CXA in the primary datacentre due to proximity and business continuity processing.

When arbitrating, the client can utilise 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 utilising distinct infrastructure, and the architecture is such that neither the A nor B feed should be expected to be 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 CXA'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 client 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 CXA and a client feed handler listening to the "A", "B", and "E" instances of a unit:

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)



1.3 Symbol Ranges, Units, and Sequence Numbers

Symbols will be separated into units, and the [symbol distribution](#) will not change intra-day. Cboe Australia does, however, **reserve the right to add multicast addresses or change the symbol distribution. Clients 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 CXA GRP port and then receiving requested messages on designated gap recovery multicast addresses. Clients 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. Clients 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. Clients will receive a total daily allowance of gap requested messages. In addition, each client 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 in section 4.5. The client can then wait until the time-based gap request limits reset or perform a spin as defined in section 1.5. If the daily allowance of gap requests is exceeded the client 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. Clients 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 clients for every gap request received by the GRP. Clients should be prepared to see replayed multicast data before or after the receipt of the gap response acknowledgement from the GRP.

[Section 4.6](#) shows an example flow of messages between a client and Cboe Australia'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 clients to connect via TCP and receive a spin of the current order book and symbols on that unit. By using the spin, a client can get the current CXA 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 client 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 client 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`, `Add Order`, and `Calculated Value` messages. `Trading Status` messages will be sent in spins for all symbols that are not "C"losed, which results in at least one message for every symbol that has not been "C"losed 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 client must buffer multicast messages received. If the `Spin Image Available` message sequence number is the client's reference point, multicast messages with larger sequence numbers should be buffered. If a non-`Spin Image Available` sequence number is the client's reference point which they send in their `Spin Request`, they should buffer from that point on. However, the client 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.

[Section 5.8](#) shows an example flow of messages between a client and CXA Multicast PITCH feed and Spin Server.

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

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

2 Protocol

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

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. **CXA reserves the right to add message types and grow the length** of any message without notice. Clients 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 7 implied decimal places (denominator = 10,000,000).
- **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 |
| <i>Hdr Length</i> | 0 | 2 | Binary | Length of entire block of messages. Includes this header and Hdr Count messages to follow. |
| <i>Hdr Count</i> | 2 | 1 | Binary | Number of messages to follow this header. |
| <i>Hdr Unit</i> | 3 | 1 | Binary | Unit that applies to messages included in this header. |
| <i>Hdr Sequence</i> | 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 CXA sends heartbeats on all real-time and gap channels with a sequence of 0 to help clients validate multicast connectivity. Heartbeats might not be sent outside of normal trading hours during scheduled maintenance.

CXA 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 client connection. **This also applies when the client is receiving a spin from the Spin Server, the heartbeats must continue to be sent from the client 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 clients 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 CXA 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 CXA book in the unit specified in the `Sequenced Unit Header`. It would be distributed in rare recovery events such as a datacentre fail-over.

| Unit Clear | | | | |
|------------------------|--------|--------|--------------|---|
| Field Name | Offset | Length | Type/(Value) | Description |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field. |
| <i>Message Type</i> | 1 | 1 | 0x97 | Unit Clear Message |
| <i>Reserved</i> | 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 summarises the `Trading Status` values in the CXA system:

- **C** = Closed. Not accepting orders or off-exchange trade reports. Implied at system start-up for all symbols.
- **A** = Pre-market. Not accepting orders, off-exchange trades may be reported.
- **T** = Trading. Continuous trading session open. Accepting orders and off-exchange trade reports.
- **M** = MOC Trading. Continuous trading session closed. Accepting only MOC orders and off-exchange trade reports.
- **P** = Post-market. MOC only trading session closed. Not accepting orders, off-exchange trades may be reported.
- **H** = Halted. Not accepting orders, only eligible off-exchange trades may be reported. Existing orders may be cancelled.
- **S** = Trading suspended. Sent in the event trading is suspended for operational reasons. Not accepting orders, only eligible off-exchange trades may be reported. Existing orders may be cancelled.

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

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

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

| Trading Status | | | | |
|-------------------------|--------|--------|----------------------|--|
| Field Name | Offset | Length | Type/(Value) | Description |
| <i>Length</i> | 0 | 1 | Binary | Length of this message including this field |
| <i>Message Type</i> | 1 | 1 | 0x3B | Trading Status message |
| <i>Timestamp</i> | 2 | 8 | Binary UTC Timestamp | Nanoseconds since epoch |
| <i>Symbol</i> | 10 | 6 | Printable ASCII | Symbol (right padded with spaces). |
| <i>Trading Status</i> | 16 | 1 | Alphanumeric | C = Closed A = Pre-market T = Trading M = MOC Trading P = Post-market H = Halted S = Trading suspended |
| <i>Market Id Code</i> | 17 | 4 | Alphanumeric | Market Identifier Codes: XASX = ASX Symbols CXAW = CXA Warrants CXAE = CXA ETF CXAQ = CXA QMF |
| <i>Reserved</i> | 21 | 1 | Binary | Reserved (undefined) |
| Total Length = 22 bytes | | | | |

3.3 Add Order Message

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

| Add Order | | | | |
|---------------------|--------|--------|----------------------|---|
| Field Name | Offset | Length | Type/(Value) | Description |
| <i>Length</i> | 0 | 1 | Binary | Length of this message including this field. |
| <i>Message Type</i> | 1 | 1 | 0x37 | Add Order Message |
| <i>Timestamp</i> | 2 | 8 | Binary UTC Timestamp | Nanoseconds since epoch |
| <i>Order Id</i> | 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. <u>Example conversion:</u> Base 10 – 1079067412513217551 Base 36 – 874XH1UZEHOV |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | | | | |
|--------------------------------|----|---|-----------------|---|
| <i>Side Indicator</i> | 18 | 1 | Alphanumeric | B = Buy Order S = Sell Order |
| <i>Quantity</i> | 19 | 4 | Binary | Number of shares being added to the book. For undisclosed orders, the number of shares is zero. |
| <i>Symbol</i> | 23 | 6 | Printable ASCII | Symbol (right padded with spaces). |
| <i>Price</i> | 29 | 8 | Binary Price | The display price of the order. |
| <i>PID</i> | 37 | 4 | Alphanumeric | Participant ID (right padded with spaces). Blank (spaces) if not attributed. |
| <i>Reserved</i> | 41 | 1 | Binary | Reserved (undefined) |
| Total Length = 42 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 CXA book is executed in whole or in part. The execution price equals the order price found in the original `Add Order` message or the order price in the latest `Modify Order` message referencing the *Order Id*.

| Order Executed | | | | |
|--------------------------|--------|--------|----------------------|--|
| Field Name | Offset | Length | Type/(Value) | Description |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field. |
| <i>Message Type</i> | 1 | 1 | 0x38 | <code>Order Executed Message</code> |
| <i>Timestamp</i> | 2 | 8 | Binary UTC Timestamp | Nanoseconds since epoch |
| <i>Order Id</i> | 10 | 8 | Binary | <i>Order Id</i> of a previously sent <code>Add Order</code> message that was executed. |
| <i>Executed Quantity</i> | 18 | 4 | Binary | Number of shares executed. |
| <i>Execution Id</i> | 22 | 8 | Binary | CXA generated day-unique execution identifier of this execution. <i>Execution Id</i> is also referenced in the <code>Trade Break</code> message. |
| <i>Contra Order Id</i> | 30 | 8 | Binary | <i>Order Id</i> of the contra order that matched with this order. |
| <i>Contra PID</i> | 38 | 4 | Alphanumeric | Contra Participant ID (right padded with spaces). Blank (spaces) if not attributed. |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | | | | |
|--------------------------------|----|---|--------|----------------------|
| <i>Reserved</i> | 42 | 1 | Binary | Reserved (undefined) |
| Total Length = 43 bytes | | | | |

3.4.2 Reduce Size Message

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

| Reduce Size | | | | |
|--------------------------------|--------|--------|----------------------|--|
| Field Name | Offset | Length | Type/(Value) | Description |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field |
| <i>Message Type</i> | 1 | 1 | 0x39 | <code>Reduce Size</code> Message |
| <i>Timestamp</i> | 2 | 8 | Binary UTC Timestamp | Nanoseconds since epoch |
| <i>Order Id</i> | 10 | 8 | Binary | <i>Order Id</i> of a previously sent <code>Add Order</code> message that has been reduced. |
| <i>Cancelled Quantity</i> | 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 |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field |
| <i>Message Type</i> | 1 | 1 | 0x3A | <code>Modify Order</code> Message |
| <i>Timestamp</i> | 2 | 8 | Binary UTC Timestamp | Nanoseconds since epoch |
| <i>Order Id</i> | 10 | 8 | Binary | <i>Order Id</i> of a previously sent <code>Add Order</code> message that has been modified. |
| <i>Quantity</i> | 18 | 4 | Binary | Number of shares associated with this order after this modify (may be less than the number entered). For undisclosed orders, the number of shares is zero. |
| <i>Price</i> | 22 | 8 | Binary Price | The order price after this modify. |
| <i>Reserved</i> | 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. Clients should be prepared to handle this scenario.

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

3.5 Trade Message

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

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

| Trade | | | | |
|---------------------|--------|--------|----------------------|--|
| Field Name | Offset | Length | Type/(Value) | Description |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field |
| <i>Message Type</i> | 1 | 1 | 0x3D | Trade Message |
| <i>Timestamp</i> | 2 | 8 | Binary UTC Timestamp | Nanoseconds since epoch |
| <i>Symbol</i> | 10 | 6 | Printable ASCII | Symbol (right padded with spaces). |
| <i>Quantity</i> | 16 | 4 | Binary | Incremental number of shares executed or reported. |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | | | | |
|--------------------------|----|---|--------------|--|
| <i>Price</i> | 20 | 8 | Binary Price | The price of the trade. |
| <i>Execution Id</i> | 28 | 8 | Binary | CXA generated day-unique execution identifier of this execution. <i>Execution Id</i> is also referenced in the Trade Break message. |
| <i>Order Id</i> | 36 | 8 | Binary | <i>Order Id</i> of the executed order. |
| <i>Contra Order Id</i> | 44 | 8 | Binary | <i>Order Id</i> of the contra order that matched with this order. |
| <i>PID</i> | 52 | 4 | Alphanumeric | Participant ID (right padded with spaces). Blank (spaces) if not attributed. |
| <i>Contra PID</i> | 56 | 4 | Alphanumeric | Contra Participant ID (right padded with spaces). Blank (spaces) if not attributed. |
| <i>Trade Type</i> | 60 | 1 | Alphanumeric | B = Broker Preferred Trade N = Trade resulting from normal matching logic <space> = Off-exchange trade report |
| <i>Trade Designation</i> | 61 | 1 | Alphanumeric | C = CXAC (Limit) P = CXAP (Mid-Point) N = CXAN (Near Point) F = CXAF (Far Point) M = CXAM (MOC) B = CXAB (BIDS Block Size) I = CXAI (BIDS Price Improved) Valid only for on-exchange executions, space otherwise. |
| <i>Trade Report Type</i> | 62 | 1 | Alphanumeric | B = Block Trade P = Large Portfolio Trade T = Large Principal Transaction S = Trades with Price Improvement L = Permitted Trade During Post Trading Hours Period M = Permitted Trade During Pre-Trading Hours Period E = Out of Hours Trade F = ETF Trade Report for unit creations or redemptions Valid only for off-exchange trade reports, space otherwise. |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | | | | |
|--------------------------------|----|---|----------------------|---|
| <i>Trade Transaction Time</i> | 63 | 8 | Binary UTC Timestamp | Nanosecond timestamp of the off-exchange trade as specified in the Trade Report submitted by the trading participant. Valid only for off-exchange trade reports, zero otherwise. |
| <i>Reserved</i> | 71 | 1 | Binary | Reserved (undefined) |
| Total Length = 72 bytes | | | | |

3.6 Trade Break Message

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

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

3.7 Calculated Value Message

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

| Calculated Value | | | | |
|---------------------|--------|--------|----------------------|--|
| Field Name | Offset | Length | Type/(Value) | Description |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field |
| <i>Message Type</i> | 1 | 1 | 0xE3 | Calculated Value Message |
| <i>Timestamp</i> | 2 | 8 | Binary UTC Timestamp | Nanoseconds since epoch |
| <i>Symbol</i> | 10 | 6 | Printable ASCII | Symbol (right padded with spaces). |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | | | | |
|--------------------------------|----|---|-------------------------|---|
| <i>Value Category</i> | 16 | 1 | Alphanumeric | 1 = Closing price 2 = iNAV values (ETF) 3 = Index values 4 = EOD NAV from issuer |
| <i>Value</i> | 17 | 8 | Binary Price | The calculated value. |
| <i>Value Timestamp</i> | 25 | 8 | Binary UTC Timestamp | Timestamp when the calculated value was generated in nanoseconds since epoch. |
| Total Length = 33 bytes | | | | |

3.8 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 |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field |
| <i>Message Type</i> | 1 | 1 | 0x2D | <code>End of Session Message</code> |
| <i>Reserved</i> | 2 | 4 | Binary | Reserved (undefined) |
| Total Length = 6 bytes | | | | |

4 Gap Request Proxy Messages

The following messages are used for initialising a TCP/IP connection to the Gap Request Proxy (“GRP”) and to request message retransmissions. Clients 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 in Section 2.4. The following messages will not be delivered using multicast.

Clients 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 client 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 |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field. |
| <i>Message Type</i> | 1 | 1 | 0x01 | Login Message |
| <i>SessionSubId</i> | 2 | 4 | Alphanumeric | <i>SessionSubId</i> supplied by CXA. |
| <i>Username</i> | 6 | 4 | Alphanumeric | <i>Username</i> supplied by CXA. |
| <i>Filler</i> | 10 | 2 | Alphanumeric | (space filled) |
| <i>Password</i> | 12 | 10 | Alphanumeric | <i>Password</i> supplied by CXA. |
| Total Length = 22 bytes | | | | |

4.2 Login Response

The `Login Response` message is sent by the GRP to the client 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 |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field. |
| <i>Message Type</i> | 1 | 1 | 0x02 | Login Response Message |
| <i>Status</i> | 2 | 1 | Alphanumeric | Accepted or reason for reject. |
| Total Length = 3 bytes | | | | |

| Login Response – Status Codes | |
|-------------------------------|--|
| ‘A’ | Login Accepted |
| ‘N’ | Not authorised (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 client's connection to the GRP server alive. Heartbeat messages are sent using the `Sequenced Unit Header` as described in sections 2.4 and 2.5.

4.4 Gap Request

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

| Gap Request | | | | |
|------------------------|--------|--------|------------|--|
| Field | Offset | Length | Value/Type | Description |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field. |
| <i>Message Type</i> | 1 | 1 | 0x03 | <code>Gap Request Message</code> |
| <i>Unit</i> | 2 | 1 | Binary | <i>Unit</i> that the gap is requested for. |
| <i>Sequence</i> | 3 | 4 | Binary | <i>Sequence</i> of first message (Lowest sequence in range). |
| <i>Count</i> | 7 | 2 | Binary | <i>Count</i> 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 |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field. |
| <i>Message Type</i> | 1 | 1 | 0x04 | Gap Response Message. |
| <i>Unit</i> | 2 | 1 | Binary | <i>Unit</i> the gap was requested for. |
| <i>Sequence</i> | 3 | 4 | Binary | <i>Sequence</i> of first message in request. |
| <i>Count</i> | 7 | 2 | Binary | <i>Count</i> of messages requested. |
| <i>Status</i> | 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 client and CXA's Multicast PITCH feed, Gap Request Proxy, and Gap Server.

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

At time 1 the client 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 client is now successfully logged into the GRP and able to request gaps. Note this is just for example purposes and in practice the client is encouraged to log into the GRP at the start of the trading day.

At time 3 and 4, the client receives sequences 310171 and 310172. These messages are in sequence and the client 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 client 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.

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

At time 8, the client 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 client 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 client 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 client receives sequence 310173 from the Gap Server. Since the last sequence applied was 310172, the client 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 client 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.

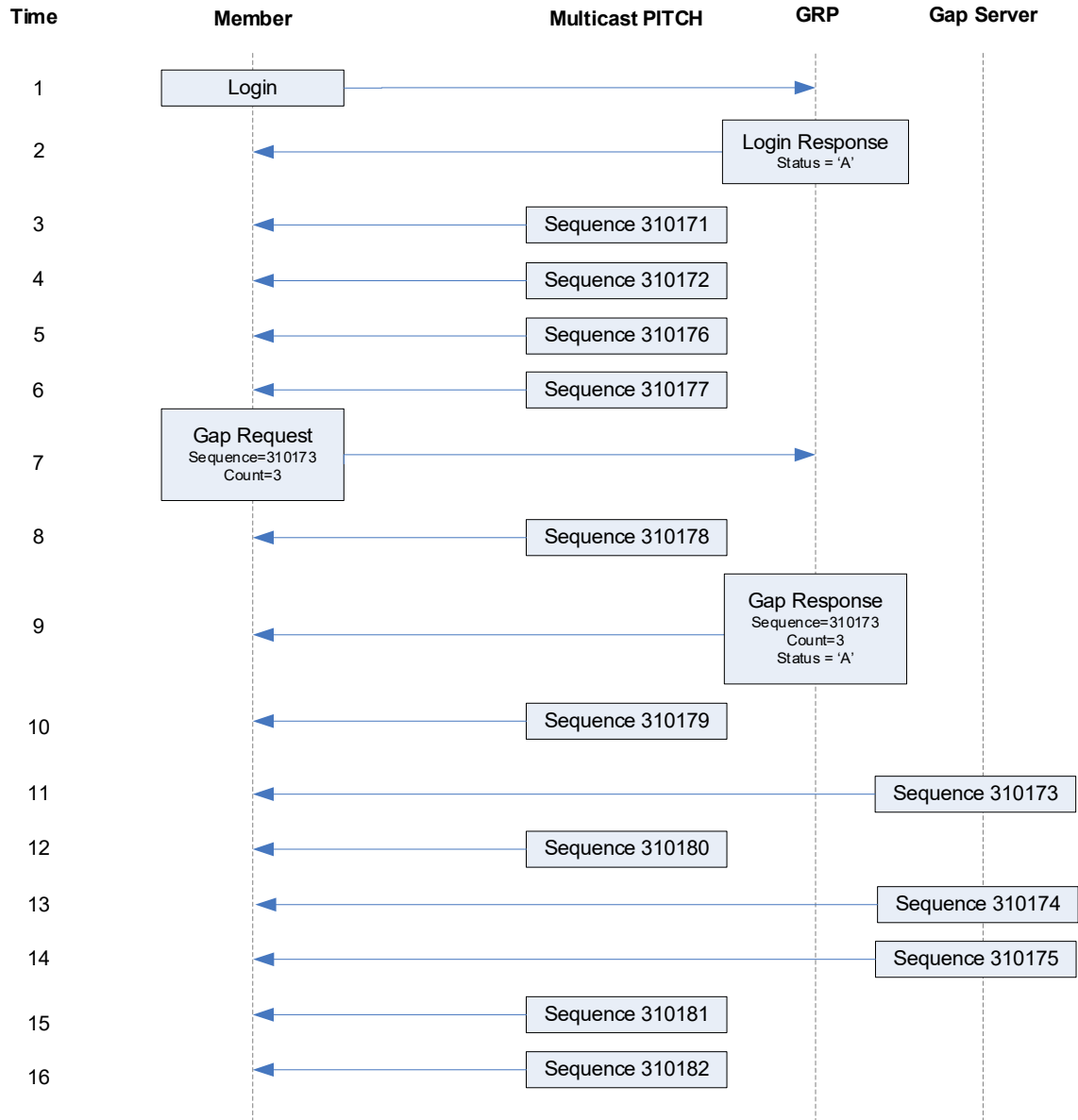
At time 13 and 14, the client receives sequences 310174 and 310175 from the Gap Server. Since the last sequence applied was 310173, the client can apply these messages to the book.

Now that all the missing sequences have been received from the Gap Server, the client 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 client 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 clients may also request gaps, and the clients should be prepared to ignore any message from the Gap Server it is not expecting or does not need.

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)



5 Spin Messages

5.1 Login

The `Login` message is the first message sent to the Spin Server by a client'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 described previously in [Section 4.1](#).

5.2 Login Response

The `Login Response` message is sent by the Spin Server to a client'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 in [Section 4.2](#).

5.3 Heartbeat

Heartbeat messages must be sent once every five seconds in order to keep the client's connection to the spin server alive. Heartbeat messages are sent using the `Sequenced Unit Header` as described in sections 2.4 and 2.5.

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 |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field. |
| <i>Message Type</i> | 1 | 1 | 0x80 | <code>Spin Image Available</code> Message |
| <i>Sequence</i> | 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 client's process to request transmission of a spin of the unit's order book. Refer to [Section 1.5](#) for more complete details regarding *Sequence* specification as well as buffering requirements.

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

5.6 Spin Response

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

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

| Spin Response - Status Codes | |
|------------------------------|--|
| 'A' | Accepted |
| 'O' | Out of Range (<i>Sequence</i> requested is greater than <i>Sequence</i> 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 client's copy of the book to make it current.

| Spin Finished | | | | |
|---------------------|--------|--------|--------------|---|
| Field Name | Offset | Length | Type/(Value) | Description |
| <i>Length</i> | 0 | 1 | Binary | <i>Length</i> of this message including this field. |
| <i>Message Type</i> | 1 | 1 | 0x83 | Spin Finished Message |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | | | | |
|------------------------|---|---|--------|---|
| 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 client and CXA's Multicast PITCH feed and Spin Server. The spin will consist of Trading Status, Calculated Value, and Add Order messages.

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

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

At time 7, the client 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 client does not have all messages cached after 310169 (they are missing 310170 and 310171), so this spin is not useful to the client.

At time 10, the client 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 client has all messages after 310175 cached.

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

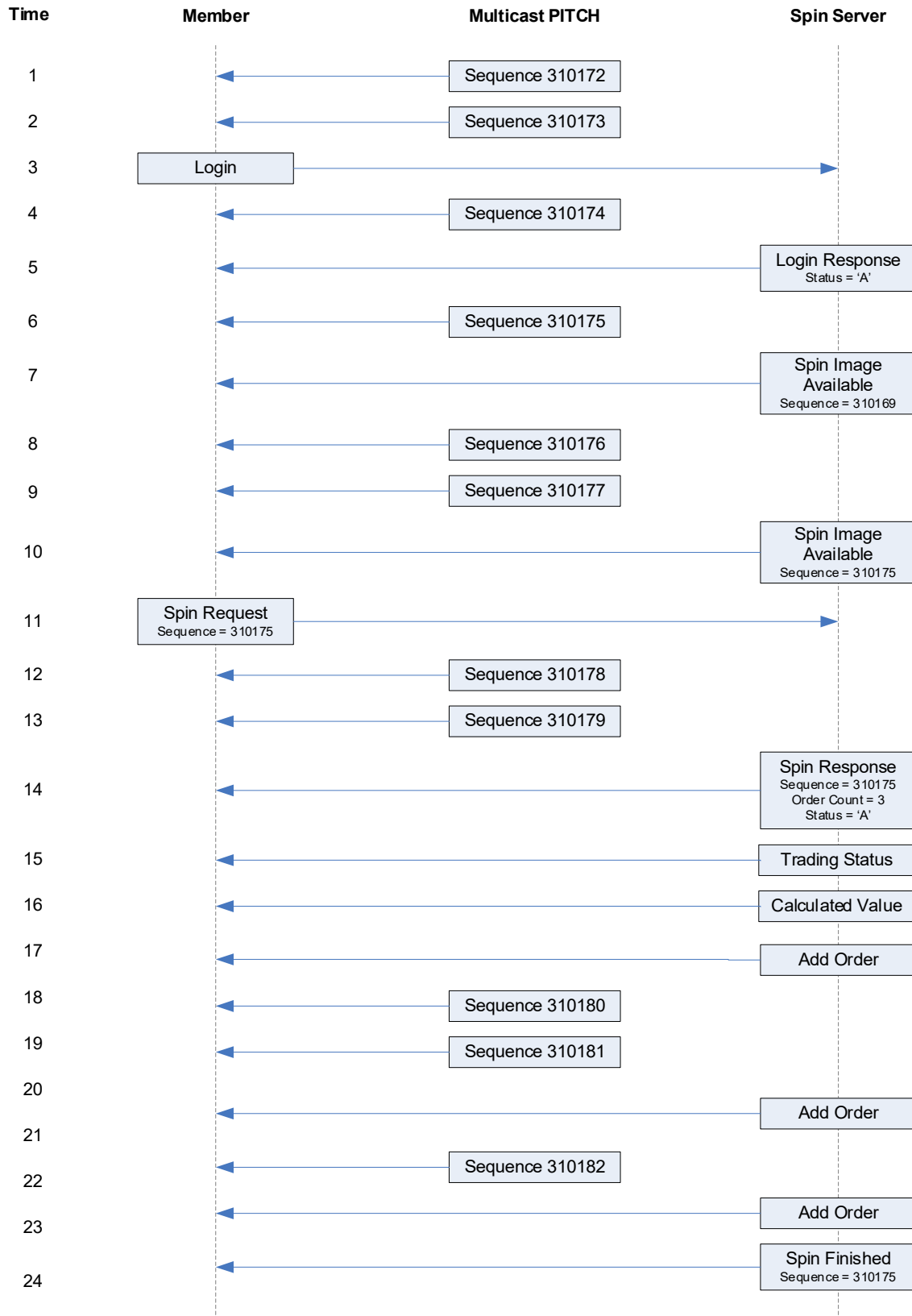
At time 14, the spin server acknowledges the spin request and indicates that three open orders will be sent.

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

Notes:

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

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)



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 |
| 0x3B | Trading Status |
| 0x37 | Add Order |
| 0x38 | Order Executed |
| 0x39 | Reduce Size |
| 0x3A | Modify Order |
| 0x3C | Delete Order |
| 0x3D | Trade |
| 0x3E | Trade Break |
| 0xE3 | Calculated Value |
| 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 in Section 2.4. 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 | "ABCD00 " |
| | 20 20 | |

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) |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

7.1.6 Trading Status

| | | |
|----------------|-------------------------|------------------------------------|
| Length | 16 | 22 bytes |
| Type | 3B | Trading Status |
| Timestamp | F0 77 BB CE 2A 6A 62 16 | 1612968348641622000 ns since epoch |
| Symbol | 5A 56 5A 54 20 20 | "ZVZT " |
| Trading Status | 54 | T = Trading |
| Market Id Code | 41 55 53 20 | "XASX" |
| Reserved | 00 | (Reserved) |

7.1.7 Add Order

| | | |
|----------------|-------------------------|------------------------------------|
| Length | 2A | 42 bytes |
| Type | 37 | 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 | 5A 56 5A 54 20 20 | "ZVZT " |
| Price | 15 CD 5B 07 00 00 00 00 | 12.3456789 |
| PID | 31 32 33 34 | "1234" |
| Reserved | 00 | (Reserved) |

7.1.8 Order Executed

| | | |
|-------------------|-------------------------|------------------------------------|
| Length | 2B | 43 bytes |
| Type | 38 | 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 | 06 40 5B 77 8F 56 1D 0B | 631WC4000006 (base36) |
| Contra PID | 35 36 37 38 | "5678" |
| Reserved | 00 | (Reserved) |

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 | BC 02 00 00 | 700 shares |
| Quantity | | |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

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 | 15 CD 5B 07 00 00 00 00 | 12.3456789 |
| 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 (On-Exchange Electronic Execution)

| | | |
|-------------------|-------------------------|--------------------------------------|
| Length | 48 | 72 bytes |
| Type | 3D | Trade |
| Timestamp | F0 77 BB CE 2A 6A 62 16 | 1612968348641622000 ns since epoch |
| Symbol | 5A 56 5A 54 20 20 | "ZVZT " |
| Quantity | BC 02 00 00 | 700 shares |
| Price | 15 CD 5B 07 00 00 00 00 | 12.3456789 |
| 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 | 06 40 5B 77 8F 56 1D 0B | 631WC4000006 (base36) |
| PID | 31 32 33 34 | "1234" |
| Contra PID | 35 36 37 38 | "5678" |
| Trade Type | 4E | N = Trade from normal matching logic |
| Trade Designation | 43 | C = CXAC (Limit) |
| Trade Report Type | 20 | " "(space) |
| Trade Transaction | 00 00 00 00 00 00 00 00 | zero |
| Time | | |
| Reserved | 00 | (reserved) |

7.1.13 Trade (Off-Exchange Trade Report)

| | | |
|--------------|-------------------------|------------------------------------|
| Length | 48 | 72 bytes |
| Type | 3D | Trade |
| Timestamp | F0 77 BB CE 2A 6A 62 16 | 1612968348641622000 ns since epoch |
| Symbol | 5A 56 5A 54 20 20 | "ZVZT " |
| Quantity | BC 02 00 00 | 700 shares |
| Price | 15 CD 5B 07 00 00 00 00 | 12.3456789 |
| Execution Id | 34 2B 46 E0 BB 00 00 00 | 0AAP09VEC (base36) |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | | |
|------------------------|-------------------------|------------------------------------|
| Order Id | 05 40 5B 77 8F 56 1D 0B | 631WC4000005 (base36) |
| Contra Order Id | 06 40 5B 77 8F 56 1D 0B | 631WC4000006 (base36) |
| PID | 31 32 33 34 | "1234" |
| Contra PID | 20 20 20 20 | Unattributed - all spaces |
| Trade Type | 20 | " " (off-exchange) |
| Trade Designation | 20 | " " (space) |
| Trade Report Type | 50 | P = Large Portfolio Trade |
| Trade Transaction Time | F0 77 BB CE 2A 6A 62 16 | 1612968348641622000 ns since epoch |
| Reserved | 00 | (reserved) |

7.1.14 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) |

7.1.15 Calculated Value

| | | |
|-----------------|-------------------------|------------------------------------|
| Length | 21 | 33 bytes |
| Type | E3 | Calculated Value |
| Timestamp | F0 77 BB CE 2A 6A 62 16 | 1612968348641622000 ns since epoch |
| Symbol | 5A 56 5A 54 20 20 | "ZVZT " |
| Value Category | 31 | 1 = Closing price |
| Value | 15 CD 5B 07 00 00 00 00 | 12.3456789 |
| Value Timestamp | F0 77 BB CE 2A 6A 62 16 | 1612968348641622000 ns since epoch |

7.1.16 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 CXA Multicast PITCH messages sent in response to various orders entered in CXA.

7.2.1 Modify Order Example

| Action | Message Description |
|--|--|
| A visible order to buy 100 ZVZT shares 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: 37 (Add Order) Timestamp: 1612968348641622000 (nanos) Order Id (base36): 100000000001 Side: B (Buy) Quantity: 100 Symbol: "ZVZT" Price: 10.00 PID: "1234" 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: 100 Price: 11.00 Reserved: 0 (Reserved) |

7.2.2 Undisclosed Order Execution Example

| Action | Message Description |
|---|--|
| An undisclosed order to buy 200 ZVZT shares at 10.00 is entered. The order is assigned Id 100000000002 and rests on the book. An Add Order message with price of 10.00 and zero quantity is sent. | Type: 37 (Add Order) Timestamp: 1612968348641622000 (nanos) Order Id (base36): 100000000002 Side: B (Buy) Quantity: 0 (undisclosed) Symbol: "ZVZT" Price: 10.00 PID: "1234" Reserved: 0 (Reserved) |
| A visible order to sell 100 ZVZT shares at 10.00 is entered. This order is executed against the resting undisclosed buy order. Since the resting order is undisclosed a Trade message is sent that includes the quantity traded on the undisclosed order. | Type: 3D (Trade) Timestamp: 1612968348641623000 (nanos) Symbol: "ZVZT" Quantity: 100 Price: 10.00 Execution Id (base36): 01000000A Order Id (base36): 100000000002 Contra Order Id (base36): 100000000003 PID: "1234" Contra PID: "5678" Trade Type: N (Normal) Trade Designation: C (Limit) Trade Report Type: <space> Trade Transaction Time: 0 Reserved: 0 (Reserved) |
| Another visible order to sell 100 ZVZT shares at 10.00 is entered. This order is executed against the resting undisclosed buy order. Since the resting order is undisclosed a Trade message is sent. | Type: 3D (Trade) Timestamp: 1612968348641624000 (nanos) Symbol: "ZVZT" Quantity: 100 Price: 10.00 Execution Id (base36): 01000000B Order Id (base36): 100000000002 Contra Order Id (base36): 100000000004 PID: "1234" Contra PID: "9123" Trade Type: N (Normal) Trade Designation: C (Limit) Trade Report Type: <space> Trade Transaction Time: 0 Reserved: 0 (Reserved) |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | |
|---|--|
| The undisclosed order is fully filled and is removed from the book. A Delete Order message is sent. | Type: 3D (Delete Order) Timestamp: 1612968348641624000 (nanos) Order Id (base36): 100000000002 |
|---|--|

7.2.3 Iceberg Order Execution Example

| Action | Message Description |
|--|--|
| An iceberg order to buy 100 ZVZT shares at 10.00 is entered, with a display quantity of 50. The order is assigned Id 100000000005 and rests on the book. An Add Order message with price of 10.00 and quantity of 50. | Type: 37 (Add Order) Timestamp: 1612968348641622000 (nanos) Order Id (base36): 100000000005 Side: B (Buy) Quantity: 50 Symbol: "ZVZT " Price: 10.00 PID: "1234" Reserved: 0 (Reserved) |
| A visible order to sell 20 ZVZT shares 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 20. The remaining visible quantity of the iceberg order is 30, with the total quantity of 80. | Type: 38 (Order Executed) Timestamp: 1612968348641623000 (nanos) Order Id (base36): 100000000005 Executed Quantity: 20 Execution Id (base36): 01000000C Contra Order Id (base36): 100000000006 Contra PID: "5678" Reserved: 0 (Reserved) |
| Another visible order to sell 80 ZVZT shares at 10.00 is entered. This order is executed against the resting undisclosed buy order. An Order Executed message is sent for the remaining visible quantity of 30. At this point the visible quantity of the order is exhausted and participants would remove the order from their books. | Type: 38 (Order Executed) Timestamp: 1612968348641624000 (nanos) Order Id (base36): 100000000005 Executed Quantity: 30 Execution Id (base36): 01000000D Contra Order Id (base36): 100000000007 Contra PID: "9123" Reserved: 0 (Reserved) |
| A Trade message is sent of the hidden quantity of the iceberg order. | Type: 3D (Trade) Timestamp: 1612968348641625000 (nanos) Symbol: "ZVZT " Quantity: 50 Price: 10.00 Execution Id (base36): 01000000E Order Id: 100000000008 (obfuscated) Contra Order Id (base36): 100000000007 PID: "1234" Contra PID: "9123" Trade Type: N (Normal) Trade Designation: C (Limit) Trade Report Type: <space> Trade Transaction Time: 0 Reserved: 0 (Reserved) |

7.2.4 Iceberg Order Replenished Example

| Action | Message Description |
|--|--|
| An iceberg order to buy 75 ZVZT shares at 10.00 is entered, with a display quantity of 50. The order is assigned Id 100000000009 and rests on the book. An Add Order message with price of 10.00 and quantity of 50. | Type: 37 (Add Order) Timestamp: 1612968348641622000 (nanos) Order Id (base36): 100000000009 Side: B (Buy) Quantity: 50 Symbol: "ZVZT " Price: 10.00 PID: "1234" Reserved: 0 (Reserved) |
| A visible order to sell 50 ZVZT shares 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 | Type: 38 (Order Executed) Timestamp: 1612968348641623000 (nanos) Order Id (base36): 100000000009 Executed Quantity: 50 Execution Id (base36): 01000000F |

Cboe Australia
Multicast Depth of Book (PITCH) Specification (Version 1.0.6)

| | |
|--|--|
| quantity of 50. At this point the visible quantity of the order is exhausted and participants would remove the order from their books. | Contra Order Id (base36): 10000000000A Contra PID: "5678" Reserved: 0 (Reserved) |
| The iceberg order is replenished with the remaining quantity of 25 shares. An Add Order is sent with an obfuscated (new) order Id. | Type: 37 (Add Order) Timestamp: 1612968348641624000 (nanos) Order Id (base36): 10000000000B (obfuscated) Side: B (Buy) Quantity: 25 Symbol: "ZVZT " Price: 10.00 PID: "1234" 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. CXA 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 | CXA will send UDP messages up to 1500 bytes. Clients 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 | Clients' 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 CXA symbol distribution across units.

| Unit | Symbol Range |
|------|---------------|
| 1 | Zero – M~~~~~ |
| 2 | N – Z~~~~~ |

Note – CXA 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 CXA Multicast Routing Parameters

| Data Centre | Rendezvous Point |
|------------------------------|------------------|
| Primary Data Centre A feed | 74.115.128.10/32 |
| Primary Data Centre B feed | 74.115.128.11/32 |
| Secondary Data Centre E feed | 74.115.128.13/32 |

For additional information about physical connectivity, refer to the [CXA Connectivity Manual](#).

8.1.4 CXA Address/Unit Distribution

The following tables describe the unit distribution across the CXA PITCH feeds.

| Primary Data Centre | | Gig-Shaped “A” Feed [AAM] 170.137.217.64/28 | | Gig-Shaped “B” Feed [ABM] 170.137.217.80/28 | |
|---------------------|---------|--|----------------|--|----------------|
| Unit | IP Port | Real-time MC | Gap Resp. MC | Real-time MC | Gap Resp. MC |
| 1 | 30501 | 233.218.133.80 | 233.218.133.81 | 233.218.133.96 | 233.218.133.97 |
| 2 | 30502 | | | | |

Note – CXA 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 Centre | | Gig-Shaped “E” Feed [AEM] 170.137.214.16/28 | |
|-----------------------|---------|--|-----------------|
| Unit | IP Port | Real-time MC | Gap Resp. MC |
| 1 | 31501 | 233.218.133.112 | 233.218.133.113 |
| 2 | 31502 | | |

Note – CXA 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 CXA symbol distribution across units.

| Unit | Symbol Range |
|------|----------------|
| 1 | Zero – M ~~~~~ |
| 2 | N – Z ~~~~~ |

Note – CXA 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 Centre | Rendezvous Point |
|-----------------------------------|------------------|
| Primary Data Centre feed | 74.115.128.12/32 |

8.2.3 CXA Address/Unit Distribution

The following tables describe the unit distribution across the certification CXA PITCH feeds.

| Primary Data Centre | | CertFeed [Cert] 170.137.217.16/28 | |
|---------------------|---------|--------------------------------------|-----------------|
| Unit | IP Port | Real-time MC | Gap Resp. MC |
| 1 | 32501 | 233.218.133.104 | 233.218.133.105 |
| 2 | 32502 | | |

Note – CXA 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

CXA may certify a number of carriers to redistribute Multicast data feeds, as defined in the [CXA Connectivity Manual](#). For more information on receiving CXA 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. CXA will use 90% of these respective bandwidths for Multicast PITCH to allow clients to use the same physical connection for order entry if desired.

10 Support

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

Revision History

| Document Version | Date | Description |
|------------------|----------|--|
| 1.0.0 | 08/04/22 | Initial version. |
| 1.0.1 | 27/06/22 | Added feed names to PITCH feed descriptions. Updated <i>Market Id Code</i> : 'XASX' = ASX Symbols |
| 1.0.2 | 15/08/22 | Updated Market Id Code in example messages. |
| 1.0.3 | 26/08/22 | Updated symbol distribution ranges to be simple alpha ranges. |
| 1.0.4 | 31/08/22 | Populated PITCH feed addresses in section 8. |
| 1.0.5 | 01/09/22 | Updated symbol range. |
| 1.0.6 | 07/11/22 | Updated <i>Trade Designation</i> values “B” and “I” for BIDS MIC codes. |