# OPENBOOK ULTRA CLIENT SPECIFICATION

**NYSE OPENBOOK ULTRA**

**NYSE AMERICAN OPENBOOK ULTRA**

| Version | Date |
| --- | --- |
| 2.1h | March 21, 2022 |

# PREFACE

## DOCUMENT HISTORY

| VERSION | DATE | CHANGE DESCRIPTION |
|---|---|---|
| **1.7a** | 04/28/2010 | Formatted into new template. IP addresses removed and link to IP Addresses page added |
| **1.8** | 05/21/2012 | Changed name to NYSE MKT throughout and Artwork updated throughout. |
| **1.9** | 10/07/2013 | All 228/229 messages replaced with 230/231 |
| **2.1** | 4/25/2017 | Reorganized/updated spec for clarity and rebranded to ICE/NYSE standard<br>Adopted new market name: NYSE MKT becomes NYSE American 7/24/17,<br>Change selected terminology to conform better to XDP usage<br>Bumped version to 2.1 to conform to Pillar migration<br>Added values E & L to Trading Status field of Full & Delta Update msgs<br>Symbol Index field (formerly Security Index) expanded from 2 to 4 bytes |
| **2.1a** | 6/22/2017 | Corrected Symbol Index field in Symbol Mapping Response msg to 4 bytes |
| **2.1b** | 02/08/2018 | Spec changes only. No change to the feed.<br>Removed remarks on the transition from v1.9 to v2.1<br>Updated section 3.2: Delta updates cannot span packets |
| **2.1c** | 10/25/2019 | Clarifications with NYSE Pillar migration.<br>• 'Quote Condition' field on full and delta messages - 'W' is no longer applicable.<br>• 'Trading Status' field on full and delta messages - 'E' is applicable for NYSE B/C symbols |
| **2.1d** | 10/15/2021 | Corrected public URLs in Reference section and Section 8. Legacy servers and addresses were decommissioned previously. |
| **2.1e** | 01/28/2022 | Additional Status of "B" in Trading Status field for msgtype 230 and msgtype 231.<br>Section 3.5 - Price fields in all message types modified to be signed binary integer values.<br>References - Updated link to IP address sheet. |
| **2.1f** | 02/18/2022 | Section 8 - Updated filenames for the Symbol Index mapping files - no content change. |
| **2.1g** | 03/11/2022 | Section 8 - Updated Symbol Mapping filename to only include the trade date - no content change. |
| **2.1h** | 03/21/2022 | Updated NYSE re-branding and document formats. No content changes. |

## REFERENCE MATERIAL

The following lists the associated documents, which either should be read in conjunction with this document, or which provide other relevant information for the user:

- ICE Global Network
- NYSE Symbology

## CONTACT INFORMATION

For technical support:

- Telephone: +1 212 383 3640 (International)
- NYSE Support Email: connectivity@nyse.com

## FURTHER INFORMATION

- OpenBook Ultra product page
- Capacity Statistics
- IP Address Spreadsheet

## Contents

# 1. OPENBOOK ULTRA

OpenBook Ultra is a price level depth of book feed which includes all displayed limit orders, trading floor interest and DMM interest. For each buy and sell price point, the format provides aggregated limit-order volume and number of orders. The feed is updated in real time as events occur.

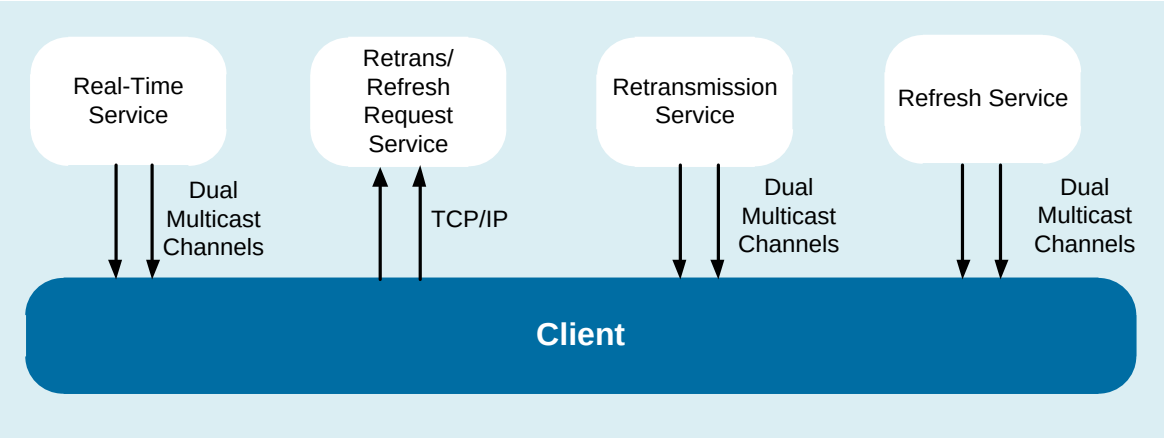
## 1.1 TERMINOLOGY AND BASICS

OpenBook Ultra data is published in the form of messages with fixed length fields. All fields are binary except a very small number that are in ASCII format. For efficient use of the network, the messages are bundled into application packets, and the packets are published via the multicast protocol. All messages in a single packet will be of the same message type.

For capacity reasons, packets are routed over a number of predefined data sets called channels. Each channel is duplicated and published to two distinct multicast groups for redundancy. The two redundant multicast groups per channel (called lines) are referred to as line A and line B. The union of the data in all channels that make up a product is called a feed.

The IP addresses and port numbers of the production and test channels for OpenBook Ultra can be found at https://www.nyse.com/publicdocs/nyse/data/IP_Addresses.xlsx. A client application receives a product by subscribing to some or all of the channels that make up the feed.

## 1.2 RECOVERING FROM ERRORS




- In case of dropped multicast packets, the client can connect to a Request Server via TCP/IP to request retransmissions of missed messages.
- In case of client late start or intraday failure, the client can connect to the Request Server and request snapshot refreshes of the state of the market.
- At system startup, each channel publishes referential data about all symbols published on the channel. If a client process misses this initial spin of symbol data, he can connect to the Request Server and request a refresh of some or all of the missed data.

In response to these requests, retransmission and refresh data is published by the exchange over dedicated multicast channels which correspond one-to-one with the real-time channels.

See Error Handling and the Request Server for complete information.

# 2. PACKETS AND HEARTBEATS

## 2.1 PACKET HEADER

All packets published by OpenBook Ultra start with a Packet Header. The Packet Header is followed by 0 or more messages, all of the same message type. (Heartbeat packets are an exception, since they do not contain any messages).

The maximum length of a packet is 1400 bytes, so no message can be longer than 1400 – 16 bytes (max packet size - the length of the Packet Header).

### 2.1.1 OpenBook Ultra Packet Header Structure

| FIELD | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **PktSize** *(prev MsgSize)* | 0 | 2 | Binary | The size of the packet in bytes, excluding these two bytes: |
| **MsgType** | 2 | 2 | Binary | This field identifies the type of message<br>• 1 – Sequence Number Reset<br>• 2 – Heartbeat Message<br>• 5 – Message Unavailable<br>• 10 – Request Response message<br>• 19 – Heartbeat Subscription message<br>• 20 – Retransmission Request Message<br>• 22 – Refresh Request Message<br>• 24 – Heartbeat Response Message<br>• 27 – Extended Refresh Request<br>• 34 – Symbol Index Mapping Request<br>• 35 – Symbol Index Mapping Refresh Message<br>• **230 – OpenBook Full Update Message**<br>• **231 – OpenBook Delta Update Message** |
| **PktSeqNum** | 4 | 4 | Binary | The packet sequence number. Incremented by 1 for each packet published in this channel except heartbeat packets. |
| **SendTime** | 8 | 4 | Binary | The time this packet was published, in milliseconds since midnight. |
| **ProductID** | 12 | 1 | Binary | Values:<br>• 12 – NYSE OpenBook Ultra<br>• 62 – NYSE American OpenBook Ultra |

| FIELD | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **RetransFlag** | 13 | 1 | Binary | Indicates whether this packet contains original, retransmitted, or refresh message. Valid values:<br>• 1 – Original packet<br>• 2 – Retransmitted packet<br>• 5 – Refresh packet with more to come<br>• 6 – Last Refresh packet in a sequence |
| **NumMsgs** | 14 | 1 | Binary | The number of messages following this header in the packet. |
| **LinkFlag** | 15 | 1 | Binary | The sequence number of this packet in a refresh sequence.<br>If RetransFlag is not = 5 or 6, this field is set to 0. |

## 2.2 HEARTBEATS

Application heartbeats are sent once a second by the real-time multicast servers (data, refresh and retransmissions channels), and once a minute by the Request Server over TCP.

A heartbeat consists of a packet containing a Packet Header and no messages. Packet Header fields values are PktSize = 14, MsgType = 2, RetransFlag = 1, and NumMsgs = 0.

The Packet Sequence Number of a heartbeat packet is the next expected sequence number, but unlike all other packets, a heartbeat does not increment the next expected sequence number. See Sequence Numbers.

Heartbeats sent by the Request Server over TCP must be acknowledged by the client. See Request Server.

# 3. MESSAGE FIELD CONTENT

Every datagram published by OpenBook Ultra consists of a Packet Header followed by 0 or more Messages, all of the same message type. Messages are contiguous data structures consisting of fixed-length fields. No names or tags appear in the message.

- All the fields are contiguous, with reserved fields for alignment issues.
- Binary fields are published in Big-Endian ordering
- All timestamps are in Eastern Standard Time (EST) or Eastern Daylight Time (EDT)
- All ASCII string fields are left aligned and null padded

## 3.1 FULL UPDATE MESSAGES

A full update of a symbol can span multiple Full Update Messages and/or multicast packets. In this case, all fields not part of the price point section will be repeated for each packet. To determine the number of price point in any given message, use the following formula:

*NumPricePoints = (SizeOfMessage – sum(size of fixed fields of message) ) / sum( size of fields for a price-point)*

Full Update Messages that span multiple packets/messages must be processed as one complete message.

For Full Update Messages that span multiple packets, if a packet is lost, then the whole message should be considered lost.

Full Update Messages contain all active price points regardless of prior period activity.

## 3.2 DELTA UPDATE MESSAGES

Unlike Full Update Messages, Delta Update Messages will not span packets.

Any price point containing a 0 quantity should be removed from the book.

If no changes have occurred for a given symbol since the last publication, no Delta Update Message is generated for that symbol.

## 3.3 SEQUENCE NUMBERS

All messages conform to the line level sequencing. Each channel A, B, C, D, etc has its own sequence number. Clients can use sequence numbers to determine the following:

- Missing (gapped) messages
- Unordered messages
- Duplicate messages.

Clients should note that the message sequence number per channel might restart from one following a failure recovery. A reset sequence number message will be sent to clients via the Multicast Groups to inform of such event.

## 3.4 SYMBOLS

The stock symbols in this feed are represented in NYSE Symbology, that is, the root optionally followed by a space and a suffix.

For example, if a symbol's root is "ABC" and its suffix is "PRA", the symbol's root/suffix is represented as: "ABC PRA\0\0\0\0", where "\0" represents a null value. Between the root and the suffix there will be one space. After the suffix, null values follow to fill the 11 characters allocated for the stock symbol field.

The Full Update message (type 230) contains both the stock symbol name and a symbol index, which is a more compact unique identifier for the symbol. Symbol Indexes are unique per symbol, are uniform across all NYSE markets and feeds. A new symbol is always assigned a new Symbol Index.

In message types other than the Full Update, such as the Delta Update message (type 231), the symbol is identified only with a Symbol Index. For this reason, clients must know the mapping between symbols and symbol indexes in order to process OpenBook Ultra.

The symbol mapping is available via the following methods:

1. OpenBook Ultra Full Update Message published at start of day over the main data channels
2. OpenBook Ultra Full Update Refresh Message published (on request) over the refresh channels
3. Symbol Index Mapping Refresh Message published (on request) over the refresh channels
4. Symbol Mapping FTP File available for download for the next trading day.

### 3.5 PRICES

Prices in this datafeed are signed binary integer values and represented by two fields: a numerator and a denominator. Pillar Equities will not publish a negative price.

All price fields in the feed convey a numerator, and share a common denominator, which is represented by the Price Scale Code field in referential data (see Full Update Message Format – Message Type 230).

$$Price = \frac{Numerator}{10^{PriceScaleCode}}$$

For NYSE Pillar Equities,

a. all symbols with a price scale code of 6 will have a max price of $2,147.48
b. all symbols with a price scale code of 4 will have a max price of $214,748.364
c. all symbols with a price scale code of 3 will have a max price of $999,999.999

# 4. MESSAGE SPECIFICATIONS

## 4.1 FULL UPDATE MESSAGE – MESSAGE TYPE 230

Full Update messages are published at the start of day, when NYSE is recovering an internal system failure, or upon request via the refresh channels. This message contains the complete order book for a single symbol, with all price points, an aggregated quantity at each price point and symbol mapping information.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **MsgSize** | 0 | 2 | Binary | The number of bytes in this message including this field.<br>*MsgSize = sum(fixed fields) + number of price Points*sum (price point fields for 1 price point)* |
| **SymbolIndex** | 2 | 4 | Binary | The ID of this symbol used in other message types. |
| **SourceTime** | 6 | 4 | Binary | The time the event occurred in the matching engine, in milliseconds since midnight<br>If the trading engine event occurred at 13:12:56, 170 millisecs and 30 microsecs, this field will contain 47576170 |
| **SourceTime MicroSecs** | 10 | 2 | Binary | The number of microseconds that have elapsed within the millisecond published in the SourceTime field. If the trading engine event occurred at 13:12:56, 170 millisecs and 30 microsecs, this field will contain 30 |
| **SymbolSeqNum** | 12 | 4 | Binary | The sequence number of this message in the set of all messages for this symbol. |
| **SourceSessionID** | 16 | 1 | Binary | Unused. Ignore any content. |
| **Symbol** | 17 | 11 | ASCII | The stock symbol in NYSE Symbology (the root, optionally followed by a space and a suffix), right-padded with NULLs.<br>Example:<br>"IBM PRA\0\0\0\0" |
| **PriceScaleCode** | 28 | 1 | Binary | The number of digits after the decimal place in all prices for this symbol. |
| **QuoteCondition** | 29 | 1 | ASCII | The current quote condition for the symbol<br>▪ Space – No special quote condition |
| **TradingStatus** | 30 | 1 | ASCII | The current trading status of the equity.<br>Valid Values: |

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| | | | | • P - Pre-Opening<br>• E – Early session<br>• B - Begin Accepting Orders<br>• O – Opened (Core Session)<br>• L – Late Session<br>• X - Closed<br>• H - Halted |
| **Filler** | 31 | 1 | Binary | Reserved for future use. Ignore any content. |
| **MPV** | 32 | 2 | Binary | The minimum price variation, also known as Tick, the minimum amount by which prices can differ. |
| **The following fields represent a price point and can repeat in a message:**<br>To identify the number of price points in the message, use the formula:<br>*( MsgSize - sum(size of fixed fields) ) / size of 1 price point*<br>Note: There maybe 0 price points in a message due to internal matching engine processing. Such a message will still increment the sequence number. | | | | |
| **Price Numerator** | 0 | 4 | Binary | The price (numerator) of this price point. Use the PriceScaleCode to determine the true dollar value of the price point. |
| **Volume** | 4 | 4 | Binary | The total interest quantity at this price point |
| **NumOrders** | 8 | 2 | Binary | The number of orders at this price point |
| **Side** | 10 | 1 | ASCII | The side of the order.<br>• B – Buy<br>• S – Sell |
| **Filler** | 11 | 1 | Binary | Reserved for future use. Ignore any content. |

### 4.2 DELTA UPDATE MESSAGE FORMAT – MESSAGE TYPE 231

A Delta Update message is published in response to events that occur in the book such as interest being added, executions, cancellations and interest routed to a different market.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **MsgSize** | 0 | 2 | Binary | The number of bytes in this message including this field.<br>*MsgSize=sum(fixed fields) + number of price Points*sum (price point fields for 1 price point)* |
| **SymbolIndex** | 2 | 4 | Binary | The unique ID of this symbol in the Full Update. |
| **SourceTime** | 6 | 4 | Binary | The time the event occurred in the matching engine, in milliseconds since midnight.<br>If the trading engine event occurred at 13:12:56, 170 millisecs and 30 microsecs, this field will contain 47576170 |
| **SourceTime MicroSecs** | 10 | 2 | Binary | The number of microseconds that have elapsed within the millisecond published in the SourceTime field. If the trading engine event occurred at 13:12:56, 170 millisecs and 30 microsecs, this field will contain 30 |
| **SourceSeqNum** | 12 | 4 | Binary | The sequence number of this message in the set of all messages for this symbol. |
| **SourceSessionID** | 16 | 1 | Binary | Unused. Ignore any content. |
| **QuoteCondition** | 17 | 1 | ASCII | The current quote condition for the symbol<br>• Space – No special quote condition |
| **TradingStatus** | 18 | 1 | ASCII | The current trading status of this symbol.<br>Valid Values:<br>• P - Pre-Opening<br>• B - Begin Accepting Orders<br>• E – Early session<br>• O – Opened (Core Session)<br>• L – Late Session<br>• C - Closed<br>• H - Halted |
| **PriceScaleCode** | 19 | 1 | Binary | The number of digits after the decimal place in all prices for this symbol. |
| **The following fields represent a price point and can repeat in a message:**<br>To determine the number of price points in the message, use the formula:<br>*( MsgSize - sum(size of fixed fields) ) / size of 1 price point* | | | | |

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| Note: There maybe 0 price points in a message due to internal matching engine processing. Such a message will still increment the sequence number. | | | | |
| **PriceNumerator** | 0 | 4 | Binary | The price (numerator) of this price point. Use the PriceScaleCode to determine the true dollar value of the price point. |
| **Volume** | 4 | 4 | Binary | The total interest quantity at this price point |
| **ChgQty** | 8 | 4 | Binary | The volume of the event taking place (i.e size of the order, cancel or execution) |
| **NumOrders** | 12 | 2 | Binary | The number of orders at this price point |
| **Side** | 14 | 1 | ASCII | The side of the order<br>• 'B' – Buy<br>• 'S' – Sell |
| **ReasonCode** | 15 | 1 | ASCII | This field identifies why the volume at the price point was modified<br>• 'O' – New order/additional interest added<br>• 'C' - Cancel<br>• 'E' - Execution<br>• 'X' - Multiple events |
| **LinkID1** | 16 | 4 | Binary | Unique ID for an execution. Correlates to the Deal ID in the gateway Execution Report msg.<br>0 = this is not an execution update. |
| **LinkID2** | 20 | 4 | Binary | Unused. Ignore any content. |
| **LinkID3** | 24 | 4 | Binary | Unused. Ignore any content. |

# 5. CONTROL, REFRESH, AND RETRANSMISSION MESSAGE SPECIFICATIONS

## 5.1 SEQUENCE NUMBER RESET MESSAGE – MESSAGE TYPE 1

This message is sent to reset the Packet Sequence Number. Scenarios in which this can occur are:

- System startup
- Recovery from NYSE system failures
- Sequence number rollover: the next packet sequence number would exceed the field's maximum value

The preceding Packet Header contains a PktSize = 18, PktSeqNum = 1, RetransFlag = 1, and NumMsgs = 1.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **NextSeqNumber** | 0 | 4 | Binary | The sequence number that will follow in the next packet. Always = 2. |

## 5.2 HEARTBEAT RESPONSE MESSAGE – MESSAGE TYPE 24

Clients that remain connected to the Request server intraday must respond within 120 seconds to any heartbeat received with a single message of this type in its own packet. This response demonstrates that the TCP connection is still alive and prevents the Request Server from closing the connection.

The preceding Packet Header must contain PktSize = 34, RetransFlag = 1, and NumMsgs = 1.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **SourceID** | 0 | 20 | ASCII | The ID of the client sending this message, left justified and null padded |

## 5.3 RETRANSMISSION REQUEST MESSAGE – MESSAGE TYPE 20

This message is sent by clients to request a retransmission of missed packets. In response, a Retransmission Response message (type 10) will be sent back over the Request TCP connection, and the retransmitted packets will be delivered over the Retransmission channels.

The preceding Packet Header must contain a PktSize = 42, RetransFlag = 1, and NumMsgs = 1.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **BeginSeqNum** | 0 | 4 | Binary | The beginning sequence number of the range of packets to be retransmitted |
| **EndSeqNum** | 4 | 4 | Binary | The ending sequence number of the range of packets to be retransmitted |
| **SourceID** | 8 | 20 | ASCII | The ID of the client sending this message, left justified and null padded |

### 5.4 BOOK REFRESH REQUEST – MESSAGE TYPE 22

This message is sent by clients requesting a full book refresh.

The preceding Packet Header must contain PktSize = 50, RetransFlag = 1, and NumMsgs = 1.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **Symbol** | 0 | 16 | ASCII | A sequence of characters representing the symbol, padded with NULLs.<br>The symbol contains the root, optionally followed by a space and an optional suffix in host format for example:<br>"IBM PRA\0\0\0\0\0" |
| **SourceID** | 16 | 20 | ASCII | The ID of the client sending this message, left justified and null padded |

### 5.5 EXTENDED BOOK REFRESH REQUEST MESSAGE – MESSAGE TYPE 27

The Extended Book Refresh Request message allows the client to request a refresh by symbol index as opposed to symbol, and to request a refresh of all symbols in a channel.

The preceding Packet Header must contain PktSize = 38, RetransFlag = 1, and NumMsgs = 1.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **SourceID** | 0 | 20 | ASCII | The ID of the client sending this message, left justified and null padded |
| **SymbolIndex** | 20 | 4 | Binary | The ID (from the last Full Refresh Update) of the symbol for which information is requested.<br>0 = Requesting all symbols in this channel |
| **MsgType** | 24 | 2 | Binary | Unused. Any content will be ignored. |

### 5.6 SYMBOL INDEX MAPPING REQUEST MESSAGE – MESSAGE TYPE 34

This message is sent by clients requesting a refresh of Symbol Index mapping information only.

The preceding Packet Header must contain PktSize = 36, RetransFlag = 1, and NumMsgs = 1.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **SourceID** | 0 | 20 | ASCII | The ID of the client sending this message, left justified and null padded |
| **SymbolIndex** | 20 | 4 | Binary | The ID (from the last Full Refresh Update) of the symbol for which information is requested.<br>0 = Requesting all symbols in this channel |

## 5.7 REQUEST RESPONSE MESSAGE – MESSAGE TYPE 10

This message will be sent immediately via TCP/IP in response to the client's request for a retransmission, refresh, or Symbol Index information. The requested information will follow in the appropriate retransmission or refresh channel.

The preceding Packet Header contains a PktSize = 42, RetransFlag = 1, and NumMsgs = 1.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **SourceSeqNum** | 0 | 4 | Binary | The sequence number assigned by the client to the original request. It is returned by NYSE in this message for the client's identification purposes. |
| **SourceID** | 4 | 20 | ASCII | The ID of the client that sent the request, left justified and null padded |
| **Status** | 24 | 1 | Character | Status of the request. Valid values:<br>• A – Accepted<br>• R – Rejected |
| **RejectReason** | 25 | 1 | Character | The reason why the request was rejected.<br>Valid values:<br>• 0 – Request accepted<br>• 1 – Rejected due to permissions<br>• 2 – Invalid sequence range (eg: low > high)<br>• 3 – Exceeded max sequence range (>1,000)<br>• 4 – Exceeded max retrans requests in a day<br>• 5 – Exceeded max refresh requests in a day<br>• 6 - Rejected. Requested seqnum > 1,000,000 packets in the past. Use refresh to recover current state if necessary. |
| **Filler** | 26 | 2 | ASCII | Reserved for future use. |

### 5.8 UNAVAILABLE MESSAGE – MESSAGE TYPE 5

This message will be sent over the Retransmission multicast channels to inform the clients of unavailability of a range of messages for which they have requested a retransmission.

The preceding Packet Header contains a PktSize = 22, RetransFlag = 1, and NumMsgs = 1.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **BeginSeqNum** | 0 | 4 | Binary | The beginning sequence number of the requested range of packets to be retransmitted |
| **EndSeqNum** | 24 | 4 | Binary | The ending sequence number of the requested range of packets to be retransmitted |

### 5.9 SYMBOL INDEX MAPPING RESPONSE MESSAGE – MESSAGE TYPE 35

This message is sent by the NYSE in response to a Symbol Index Request.

Please note: When requesting Symbol Index Mapping, you need to send the request to the symbol's corresponding channel. i.e symbol ABC to Channel AA, BBB to Channel BB otherwise your request will not be honored.

| FIELD NAME | OFFSET | SIZE (BYTES) | FORMAT | DESCRIPTION |
|---|---|---|---|---|
| **Symbol** | 0 | 11 | ASCII | The stock symbol in NYSE Symbology (the root, optionally followed by a space and a suffix), right-padded with NULLs.<br>Example:<br>"IBM PRA\0\0\0\0\0" |
| **Filler** | 11 | 1 | | Ignore any content. |
| **Symbol Index** | 12 | 4 | Binary | This field identifies the numerical representation of the symbol. |

# 6. ERROR HANDLING AND THE REQUEST SERVER

NOTE: A Source ID is a string that uniquely identifies an OpenBook Ultra client. Clients will need one or more Source IDs in order to use the Request Server to recover from errors. Please contact Connectivity to order Source IDs.

## 6.1 REQUEST SERVER

The Request Server has a dedicated server IP/port pair for every channel in the OpenBook Ultra feed. For example, NYSE has 8 data channels, and there are 8 corresponding Request Server ports, one for each channel. Requests for data from a particular channel must be sent to the corresponding Request port.

It is possible to connect to the Request Server only as needed, disconnecting after each request, but it is recommended that you remain connected to all Request Server ports for the entire trading day.

The Request Server is subject to IP Table filtering in order to safeguard against events similar to denial-of-service attacks. The filtering prevents any client from making further connections to the Request Server after the client has connected a truly excessive number of times.

Once a client establishes a TCP/IP connection, the Request Server will send a heartbeat to the client approximately every 60 seconds. Clients must respond with a Heartbeat Response message within 120 seconds, otherwise the Request Server will assume the client or the network has failed and close the connection.

### 6.1.1 Request Queuing

Clients may send several requests at the same time with the same Source ID. There is no need to wait for one request to be fulfilled before requesting another one.

Responses to all requests are published in the order in which they are received, although overlapping requests may be de-duplicated for efficiency.

## 6.2 HANDLING SEQUENCE NUMBER GAPS

Since multicast is an unreliable protocol, packets can be dropped. For this reason, clients are advised to process both lines in a channel. If a Packet Sequence Number gap occurs on one line, the gap can be filled immediately from the other.

If a gap occurs on both lines simultaneously, the client can send a Retransmission Request Message (Msg Type 20) via TCP to the Request port corresponding to the channel that gapped. The Retransmission Request contains the sequence number range of the missing packets.

On receipt of a Retransmission Request message, the Request Server will send back a Request Response Message (Msg Type 10) over the TCP connection. If any of the fields of the Retransmission Request contained malformed or meaningless information, the request will be rejected. If the request is accepted, the Retransmission Server will re-send the requested packets via multicast over the Retransmission channel corresponding to the data channel that gapped.

If the request is rejected for exceeding a predefined system limit, the client may be prevented from making any further requests. See Request Quotas. If further requests are required, please contact NYSE.

### 6.2.1 Retransmission Format

Retransmitted packets have the same format and content as the originally published messages, including the Sequence Numbers, except that the Retrans Flag in the Packet Header is set to 2 (Retransmitted).

## 6.3 RECOVERING FROM CLIENT LATE STARTS OR INTRADAY FAILURES

If a client feed handler experiences a late start or an intraday failure, the client will usually want to receive snapshots of the current market state for each symbol before resuming processing of real-time data. To do this, the client requests a refresh from the Refresh Server.

Specifically, a late-starting or recovering client should

1. Subscribe to the Publisher multicast channels. Any messages received should be cached but not processed until all refresh information is processed.
2. Connect to all Request Server ports. These connections should be maintained all day.
3. Subscribe to the Refresh multicast channels
4. Send an Extended Book Refresh Request Message (Msg Type 27) to each Request Server

The Refresh Request contains the client's Source ID, and a Symbol Index, either specifying a particular symbol to be refreshed or, if 0, specifying all symbols in this channel.

On receipt of a Refresh Request message, the Request Server will send back a Request Response Message (Msg Type 10). If any of the fields of the Refresh Request contained malformed or meaningless information, the request is rejected. If the request is rejected for exceeding a predefined system limit, the client will be prevented from making any further requests. See Request Quotas. If further requests are required, please contact NYSE.

If the request is accepted, the Refresh Server will send the snapshot message(s) over the specific Refresh channel. Refresh packets have a RetransFlag of 5, except the last packet in the sequence, whose RetransFlag is 6.

All these messages should be used to rebuild the current state of the order book. Once all refresh messages are processed, messages from the Publisher can now be processed. Note that any messages received whose SymbolSequenceNumbers are lower than that received in the refresh information should be discarded.

No dedicated retransmission service is available for the Refresh Server; if a gap is detected in the Symbol SequenceNumbers in a refresh channel, clients should submit another refresh request.

## 6.4 REFRESHING SYMBOL INFORMATION

At system startup, each channel publishes a Full Update Message (Msg Type 230) for every symbol published on this channel. The Full Refresh message contains referential data for the symbol, including ticker name and Symbol Index.

If a client process misses the initial spin of Full Updates for whatever reason, he may need to receive a refresh of some or all symbol index mapping data before resuming processing of real-time data. To do this, the client can follow the procedure described in Recovering from Client Late Starts or Intraday Failures, but send a Symbol Index Mapping Request Message (Msg Type 34) to the Request Server instead of a Refresh Request Message.

## 6.5 REQUEST QUOTAS

The table below summarizes the retransmission/refresh request limitations for the OpenBook Ultra feed. The numbers below represent the thresholds per channel.

Any client who has been blocked by exceeding these quotas can contact NYSE to have their quotas reset.

| CAPABILITY | DESCRIPTION |
|---|---|
| **Prevention of invalid subscribers** | Incoming requests from subscribers that are not in the enabled subscriber's source ID list will not be honored. |
| **Limitation of number of packets per Retrans Request** | Retransmission requests for more than 1,000 packets will not be honored. |
| **Limitation of timeliness of Retrans Requests** | Retransmission requests for sequence numbers more than 1,000,000 lower than the current sequence number will not be honored. |
| **Limitation of number of Retrans Requests in a day** | Retransmission requests from a client who has already made 5,000 retransmission requests today will not be honored, and the client will be blocked from making retransmission requests for the remainder of the day. |
| **Limitation of number of Refresh Requests in a day** | Refresh requests from a client who has already made 5,000 refresh requests today will not be honored. |

# 7. OPERATIONAL INFORMATION

## 7.1 PUBLICATION PERIOD

The following section specifies the frequency and publication period for each message type disseminated by the OpenBook Ultra product.

| MESSAGE | MESSAGE TYPE | PUBLICATION PERIOD |
|---|---|---|
| **OpenBook Ultra Full Update Message** | 230 | OpenBook Full update messages are published on system startup (approximately 12:15am ET), or when NYSE is recovering from a system failure. |
| **OpenBook Ultra Delta Message** | 231 | OpenBook Delta messages are published on system startup (approximately 12:15am ET), or in real-time response to matching engine events. |

## 7.2 CHANNELIZATION

The datafeed is published over multicast channels by alphabetic symbol ranges.

Eight alpha channels for NYSE and four channels for NYSE American:

NYSE Channels

| NUM | SYMBOL RANGE |
|---|---|
| **1** | Starting with A-B |
| **2** | Starting with C |
| **3** | Starting with D-F |
| **4** | Starting with G-J |
| **5** | Starting with K-M |
| **6** | Starting with N-R |
| **7** | Starting with S-T |
| **8** | Starting with U-Z |

NYSE American Channels

| NUM | SYMBOL RANGE |
|---|---|
| **1** | Starting with A-C |
| **2** | Starting with D-J |
| **3** | Starting with K-R |
| **4** | Starting with S-Z |

## 7.3 NYSE CERT TESTING

The NYSE CERT environment is available for client testing **Monday to Friday from 8:00am to 5:00pm.**

NYSE CERT market data is published over different multicast groups than the production environment.

Customers can subscribe to proprietary market data based on automatically generated order flow OR can enter orders into Cert gateways to generate market data scenarios of their own.

Customers can reach the Technology Member Services for support via tms@nyse.com.

# 8. SYMBOL MAPPING FILE

The OpenBook Ultra symbol mapping files for each market are available on the prior day by 12 midnight US Eastern Time.

The OpenBook symbol mapping files are available at: ftp://ftp.nyse.com/OpenBook/SymbolMapping/

Filename for NYSE:

- NYSESymbolMapping_YYYYMMDD.xml

Filename for NYSE American:

- AmericanSymbolMapping_YYYYMMDD.xml

## 8.1 OPENBOOK SYMBOL MAPPING FILE FORMAT EXAMPLE

```
<?xml version='1.0' encoding='ISO-8859-1'>
    <xs:schema xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' >
        <xs:complexType name='SymbolMap' >
            <xs:sequence>
                <xs:element name='Symbol' type='xs:string' >
                </xs:element>
                <xs:element name='Index' type='xs:int' >
                </xs:element>
                <xs:element name=Channel' type='xs:int' >
                </xs:element>
                <xs:element name='ExchangeID' type='xs:int' >
                </xs:element>
            </xs:sequence>
        </xs:complexType>
        <xs:element name='SymbolMappingFile' >
            <xs:complexType>
                <xs:sequence>
                    <xs:element name='SymbolMap' type='SymbolMap' >
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:schema>
```