



XDP OPTIONS CLIENT SPECIFICATION

NYSE ARCA, NYSE AMEX

Version

1.0g

Date

18 December 2014

© 2015 NYSE. All rights reserved. No part of this material may be copied, photocopied or duplicated in any form by any means or redistributed without the prior written consent of NYSE. All third party trademarks are owned by their respective owners and are used with permission. NYSE and its affiliates do not recommend or make any representation as to possible benefits from any securities or investments, or third-party products or services. Investors should undertake their own due diligence regarding securities and investment practices. This material may contain forward-looking statements regarding NYSE and its affiliates that are based on the current beliefs and expectations of management, are subject to significant risks and uncertainties, and which may differ from actual results. NYSE does not guarantee that its products or services will result in any savings or specific outcome. All data is as of December 18, 2014. NYSE disclaims any duty to update this information.

PREFACE

DOCUMENT HISTORY

The following table provides a description of all changes to this document.

VERSION NO.	DATE	CHANGE DESCRIPTION
1.0	03/31/14	Initial publication of document
1.0a	04/23/14	Reworked explanations for streams, sequence numbers, line arbitration, refresh
1.0b	5/02/14	<p>TCP request/response formats modified for conformance with XDP Common Client Spec</p> <p>Client and server both use packet header on all request and response packets</p> <p>Heartbeats sent by server are packet headers as in multicast feeds (client HB responses continue to be HB Response msgs)</p> <p>Eliminate TCP Login Request msg (456) and TCP Login Request Response msg (457)</p> <p>Eliminate 2nd Request Response msg at end of download (Packet Header delivery flag indicates end of download)</p> <p>Corrections based on QA feedback:</p> <p>Removed Complex Correction message</p> <p>Added Stream IDs to msg type 437 and 439</p> <p>Added SourceTime field to msg type 413</p> <p>Corrected field order in 505 and 515</p> <p>Corrected msg size for msg type 437</p> <p>Corrected side format for msg type 415 and 429</p>
1.0c	5/19/14	<p>Fixes to minor typos</p> <p>Backed out changes to TCP request/response formats incorrectly made in version 1.0b. Corrected content:</p> <p>Client and server do not use packet headers over TCP connections</p> <p>Over TCP connections, Heartbeats and heartbeat responses are message type 12</p> <p>TCP Login Request msg (456) and TCP Login Request Response msg (457) are used – Source ID field corrected to 10 bytes long</p>
1.0d	6/10/14	<p>Msg type 437: added Symbol, OptionSymbol and Group ID fields to</p> <p>Msg type 439:</p> <p>Removed SourceTime, SourceTimeNS and SymbolSeqNum</p> <p>Added ComplexSymbol field</p> <p>Msg types 403 and 405: Second level offset and Third level offset fields expanded from 1 byte binary to 2 bytes binary</p> <p>Msg type 456: corrected field offsets and MsgSize</p> <p>Max packet size corrected from 1500 to 1400 everywhere</p> <p>Clarified explanatory text in sections 1.3, 1.3.3, 1.3.4, 1.5.2, 1.5.3, 1.6, 2.2.2, 5.1</p>
1.0e	6/19/14	1. Byte alignment changes for Msg type 437
1.0f	7/11/14	<p>Msg 403 and 405 (Depth): Added explanations of price levels</p> <p>Msg 439 (Complex Def): added 2-byte filler after NoOfLegs field</p>

		<p>MSg type 447: removed Reject code 7 as possible value</p> <p>Msg type 460: added new optional TCP Logout Request msg</p> <p>Msg 456: added 2-byte filler at the end of the message</p> <p>Msg type 503 and 505 (Depth Ref): Corrected length of price offsets</p> <p>Msg types 413, 427, 509, 515 (Imbalance): added O (Opening) as possible Auction Type value</p> <p>Packet header: removed 18, 19, 20 and 21 as possible Delivery flag values and any references to said throughout the document</p> <p>Indicated Series Seq Num in Refresh message would be different than that of original message</p>
1.0g	11/14/14	Rebranded the document to ICE standard

CONTENTS

1.	OVERVIEW	6
1.1	Introduction	6
1.2	Access to Market Data	6
1.3	Refresh Functionality	8
1.4	Symbol Index Mapping	9
1.5	Processing Guidelines	10
1.6	Heartbeats.....	16
1.7	Operational Information	17
2.	XDP MESSAGES AND PRODUCTS	18
2.1	Products	18
2.2	Messages.....	18
3.	OUTRIGHT OPTIONS SYMBOL MESSAGE FORMATS	21
3.1	Outright Quote Message Structure – Msg Type 401	21
3.2	Outright Market Depth Message Structure, Buy – Msg Type 403.....	22
3.3	Outright Market Depth Message Structure, Sell – Msg Type 405	23
3.4	Outright Trade Message Structure – Msg 407.....	24
3.5	Outright Trade Cancel Message Structure – Msg Type 409.....	25
3.6	Outright Trade Correction Message Structure – Msg Type 411	26
3.7	Outright Imbalance Message – Msg Type 413.....	27
3.8	Outright Crossing RFQ Message – Msg 415.....	28
3.9	Outright Summary Message Structure – Msg 417.....	29
3.10	Underlying Status Message Structure – Msg 419	30
3.11	Outright Series Status Message Structure – Msg 421.....	31
4.	COMPLEX MESSAGE FORMATS	32
4.1	Complex Quote Message Structure – Msg Type 423.....	32
4.2	Complex Trade Message Structure – Msg 425	33
4.3	Complex Imbalance Message – Msg Type 427	34
4.1	Complex Crossing RFQ Message– Msg 429	35
4.2	Complex Summary Message Structure – Msg 431	36
4.3	Complex Status Message – Msg 433.....	37
5.	INDEX MAPPING MESSAGE FORMATS	38
5.1	Underlying Index Mapping Message – Msg 435.....	38
5.2	Series Index Mapping Message– Msg 437.....	40
5.3	Complex Symbol Definition Message – Msg 439.....	42
6.	CONTROL MESSAGE FORMATS	44
6.1	Stream ID Message – Msg 455.....	44
6.2	TCP Login Request Message – Msg 456.....	44
6.3	TCP Login Request Response Message – Msg 457.....	45
6.4	TCP Heartbeat Message - Msg 12	45
6.5	Test Request Message – Msg 458.....	46
6.6	Test Request Response Message – Msg 459	46
6.7	Underlying Index Mapping Request Message (Msg Type 441).....	47
6.8	Series Index Mapping Request Message (Msg Type 443).....	48
6.9	Complex Symbol Definition Request Message (Msg Type 445)	49
6.10	Request Response Message (Msg Type 447).....	50

6.11	TCP Logout Request message - 460	51
6.12	Sequence Number Reset (Msg Type 1)	51
7.	REFRESH MESSAGE FORMATS	52
7.1	Refresh Outright Quote Message Structure – Msg Type 501	52
7.2	Refresh Outright Market Depth Message Structure, Buy – Msg Type 503	53
7.3	Refresh Outright Market Depth Message Structure, Sell – Msg Type 505	54
7.4	Refresh Outright Trade Message Structure – Msg 507	55
7.5	Refresh Outright Imbalance Message – Msg Type 509	56
7.6	Refresh Complex Quote Message Structure – Msg Type 511	57
7.8	Refresh Complex Trade Message Structure– Msg 513	58
7.9	Refresh Complex Imbalance Message – Msg Type 515	59
7.10	Refresh Outright Trade Correction Message Structure – Msg Type 517	60

1. Overview

This chapter provides detailed information about the features of the feed, to support the development of client applications by Traders, Independent Software Vendors and Quote Vendors.

1.1 INTRODUCTION

The XDP Options data feeds have the following high-level features:

- Multicast data delivery
- Reliable network solution
- Ultra-low latency
- High Availability
- High level of scalability

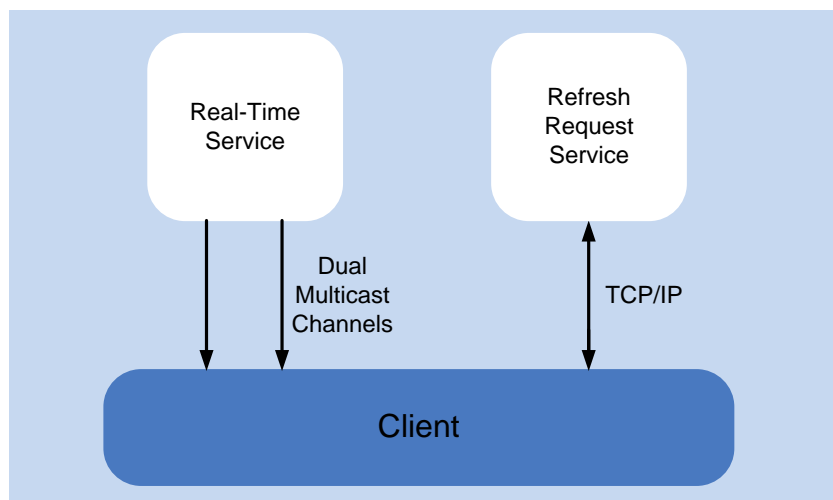
This chapter provides detailed information about the features of the feeds, to support the development of client applications by Traders, Independent Software Vendors and Quote Vendors.

The following chapters of this document provide details that are specific to each of the market data sets, including formats for each message type.

1.2 ACCESS TO MARKET DATA

Clients subscribe to multicast addresses for real-time market data and refresh messages, and may optionally connect to a TCP/IP server to request Symbol Mapping data in the event of late client start, or client failure during the trading day.

Figure 1 Access to Market Data



1.2.1 XDP Options Data Fundamentals

Real-time XDP Options data is message-based with fixed length fields (all fields are binary except a very small number of ASCII fields). In order to make efficient use of the network, the market data **Messages** are bundled into larger application **Packets**. The packets are published via the multicast protocol.

For capacity reasons, the market data packets are routed to clients via 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 (known as **Lines**) are referred to as line A and line B.

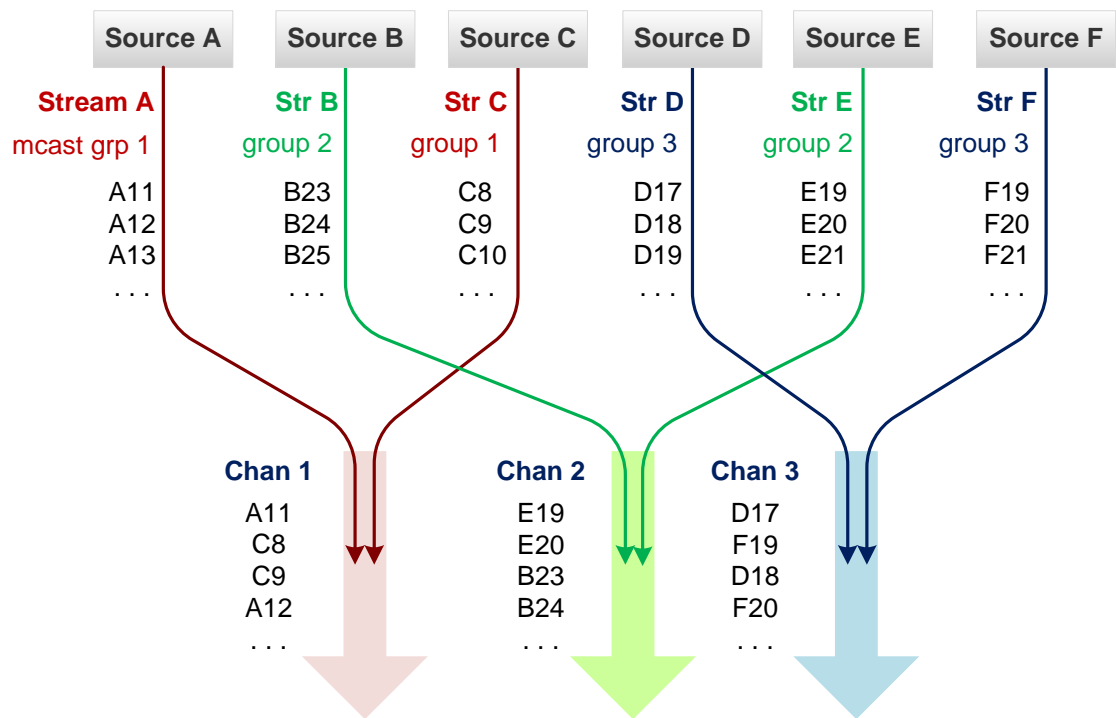
1.1.1.1 Streams and Sequence Numbers

Due to the very high volume of data in the options markets, a given channel's data typically originates from several different Matching Engine servers. The subset of a channel's data that originates from a given server is referred to as a **Stream**. Each

packet in a channel is marked with a Stream ID identifying which stream the packet belongs to, and therefore which server the packet came from.

Unlike typical market data feeds in which each channel has a single sequence number series, in XDP Options feeds, each stream within a channel has its own sequence number series. And unlike typical feeds in which a sequence number uniquely identifies a message, in XDP Options feeds, a Stream ID and a sequence number taken together uniquely identify a message.

The advantage to be gained for accepting this modest added complexity is that XDP Options data can be published to the client directly from the Matching Engine servers, with no extra tier of Market Data Publishing applications in between. This results in significantly lower latency to the client.



1.3 REFRESH FUNCTIONALITY

If a client experiences a loss of data, either because his application fails and restarts intraday, or because he experiences a sequence number gap on one or more streams, he needs to receive a snapshot of the current state of the market for all affected symbols in order to be in sync with the market again. XDP Options provides this snapshot functionality by periodically publishing Refresh messages directly in the product channels. On client restart or detection of a gap, the client simply continues listening to the product channels and processing all messages received, including the Refresh messages. Within at most two minutes, the client is guaranteed to be fully in sync with the market once again.

This inline snapshot publication makes packet request functionality and a dedicated refresh feed unnecessary.

Refresh messages are assigned a different Message Type than their standard counterparts. Furthermore, a specific value is displayed in the Delivery flag of the packet header to indicate that the packet contains at least one Refresh message.

For Integrated Top feeds, data points that automatically refresh per symbol are:

Refreshable data	Original message	Refresh message
Current quote	Quote	Refresh Quote
Last Trade	Trade, Correction, Cancel	Refresh Trade, Refresh Correction
Current imbalance, if any	Imbalance	Refresh Imbalance

For the Outright Depth of Book feed, data points that automatically refresh per symbol are:

Refreshable data	Original message	Refresh message
Current 3 best price points with aggregated volumes (bid side and ask side)	Market Depth	Refresh Market Depth

Symbol-status and Crossing RFQ messages are time-sensitive, and therefore are only published as they **occur**.

1.3.1 Integrated Top: refreshes of current Quote data

In an Integrated Top feed (Outright or Complex), whenever a symbol publishes a new Quote a 2-minute quote refresh timer is set. If another new Quote is published within the next two minutes, the quote refresh timer is set again, so it starts over again.

As long as the symbol keeps publishing quotes within 2 minutes of each other, the quote refresh timer never expires, and no Refresh Quote message is ever sent. If 2 minutes pass and no new Quote has been published, the timer expires and the information published in the current Quote message is sent again as a Refresh Quote message. The timer is then set again, and if another 2 minutes passes with no new Quote, the same Refresh Quote is sent again. This continues until a new Quote is published.

If a symbol has no current quote because it is before start of day or because the symbol is halted, Refresh Quote messages with defaulted values are sent every 2 minutes.

Note that although there are no Symbol Status Refresh messages, it is possible to determine that a symbol is halted based on the non-significant price values in the Quote messages.

1.3.2 Integrated Top: refreshes of current Imbalance data

The Refresh Imbalance algorithm involves a dedicated Imbalance Refresh timer and is exactly the same as the Refresh Quote algorithm, except that when there is no current imbalance, no Refresh Imbalance messages are sent.

1.3.3 Integrated Top: refreshes of last Trade data

The Refresh Trade algorithm involves a dedicated Trade Refresh timer and is exactly the same as the Refresh Quote algorithm, except that there is added complexity when Corrections and Cancels to the last Trade come into the picture.

If a Correction to the last Trade is published, the Trade Refresh timer is set. If 2 minutes pass and no new Trade or Correction/Cancel to the last Trade is published, the last Correction is republished as a Refresh Correction. This continues every 2 minutes as long as there is no new Trade or Correction/Cancel to the last Trade.

If a Cancel to the last Trade is published, the Trade Refresh timer is set. If 2 minutes pass and no new Trade or Correction/Cancel to the last Trade is published, the new last Trade (previously the second-to-last Trade) is republished as a Refresh Trade, since it is now the new last Trade. Note that any corrections that may have been made to the new last Trade are applied and reflected in the Refresh Trade. This continues every 2 minutes as long as there is no new Trade or Correction/Cancel to the last Trade.

1.3.4 Depth of Book: refreshes of current book data

The Refresh Depth algorithm involves two dedicated Depth Refresh timers: one for the buy side and one for the sell side. The algorithm for each timer is exactly the same as the Refresh Quote algorithm.

Note that although there are no Symbol Status Refresh messages, it is possible to determine that a symbol is halted based on the non-significant price values in the Depth messages.

1.4 SYMBOL INDEX MAPPING

To promote compact messages and therefore high throughput and low latency, underlying symbols, series symbols and complex series symbols are not published as full names in market data messages, but as symbol index codes. Mappings between all symbol index codes and the corresponding full symbol names are provided by a set of symbol index mapping messages. These mapping messages also contain key referential data about each symbol.

Symbol Index Mapping messages come in three different message types corresponding to symbol type: Underlying, Outright, and Complex.

These messages are published over multicast channels as well as by request via the Request Server.

1.4.1 Symbol Mapping Via Multicast

At the start of the trading day, three sets of symbol index mapping messages (Underlying, Outright, and Complex Index Mapping messages) are published over the multicast channels. Any given channel will publish only the mapping messages for symbols that will appear in that channel.

In the event a symbol is created during the trading day, individual Index Mapping messages can also be published intraday.

1.4.2 Symbol Download via the Request Server

The Request Server can be used to request symbol mapping downloads intraday. Clients must connect to the Request Server via a TCP/IP connection first before sending and receiving the symbol download request and response messages (ack/nack).

To establish a connection, Clients must send a TCP Login Request message and wait for a response. As long as a TCP connection is established, the server generates a Heartbeat every minute to indicate that the session is active. Clients who wish to remain connected throughout the day must respond with a Heartbeat Response message within 5 seconds, or the server will close the connection. Clients can also send a Test message at any time to verify the state of their connections.

The Symbol download request contains a Source ID. The Source ID identifies the client application, and will be supplied by the exchange. It is important to note that only one Source ID can be used per application session. Requests are made by message type (Underlying, Outright, and Complex), and can be defined to request: all symbols of a given type, all symbols of a given type for a given channel, or a specific symbol of a given type.

The Request Server responds to each download request with a Refresh Response message. If the request is accepted, the symbol download is provided over the same TCP/IP connection. When the download is complete, the Request Server generates a second Request Response to indicate that the transmission is done.

The number of refreshes allowed per client per day is limited to a total of 100 requests.

The retransmission request may be rejected for any of the following reasons:

- Invalid Source ID (username)
- Invalid Channel ID
- Incorrectly formatted request packet
- Number of requests in the current day exceeds the predefined system limit

In case of a rejection, the Request Response message indicates the reason for failure.

If the reason for failure is exceeding a predefined system limit, clients are asked to not make any further requests. If further requests are required, the client should contact NYSE Euronext.

Note that TCP communications to and from the Request Server are not compressed.

1.5 PROCESSING GUIDELINES

1.5.1 General Format Notes

The following processing notes apply to all messages:

- All fields will be sent for every message
- Only field values will appear in the published messages (for example, no names or 'tags' will appear in the message)
- The field names that appear in the message format documents are for reference purposes only
- All the fields are contiguous, with reserved fields for alignment issues
- All field sizes are fixed and constant
- The message sizes may vary due to optional trailing fields
- Binary fields are published in Little-Endian format
- ASCII fields are left-aligned and null padded
- Segmentation of messages across packets will not be supported. This means a message will never straddle a packet boundary.
- Reserved filler fields defined as ASCII are filled with a space, and those defined as Binary are populated with '0'.

1.5.2 Compression

All XDP for Options messages on multicast channels are compressed using LZ4. (Note: mapping messages sent out by the Request Server via TCP (query based) are not compressed.)

LZ4 compresses the entirety of each packet except for the first 2 bytes. These first two non-compressed bytes indicate the compressed packet size. This allows connecting applications to determine the size of the packet to read from the channel before decompressing.

The first field of the XDP Packet Header is 2 bytes long and it contains the size of compressed packet including the size of the field itself (2-bytes). Thus to determine the size of the packet to decompress, users should use the value in this field minus two.

The maximum size of a decompressed packet is 1,400 bytes.

XDP packets are compressed as a whole. Each decompressed packet will contain zero or more messages of varying lengths (a heartbeat consists of a packet header only).

1.5.3 Packet Structure

All packets of data sent on the multicast XDP feeds will have a common packet header followed by one or more messages (including the Heartbeat which consists of a packet header followed by a Stream ID message, type 455). For TCP connections to the Request Server, messages are not packetized either from the client or from the server.

The packet header format is the same for all packets, and contains packet length, number of messages within the packet, packet sequence number, and so forth.

The format of each message in the packet depends on message type, but each message will start with a message size and a message type.

The maximum length of a decompressed packet is 1,400 bytes.

The message size will never exceed the maximum packet length (less the packet header size).

PACKET HEADER	MESSAGE 1	MESSAGE 2	...	MESSAGE N
---------------	-----------	-----------	-----	-----------

The packet header provides information including the total packet length, a packet sequence number and the number of messages within the packet. The format is as follows:

Table 1 Packet Header Fields

FIELD	OFFSETS	SIZE (BYTES)	FORMAT	DESCRIPTION
PktSize	0	2	Binary	Size of the packet including the 16 -byte packet header in bytes
DeliveryFlag	2	1	Binary	A flag that indicates whether this is an original, retransmitted, or 'replayed' message. Valid values include: <ul style="list-style-type: none"> 1 – Heartbeat 2 – Contains at least one Refresh message 10 – XDP Failover 11 –Original messages 12 – Sequence Number Reset Message
NumberMsgs	3	1	Binary	Number of messages in the packet and also used to determine the next sequence number, see Sequence Numbers .
SeqNum	4	4	Binary	The message sequence number assigned by XDP for each channel. It is used for gap detection, unique for each broadcast stream (except if reset during the day), see Sequence Numbers .
SendTime	8	4	Binary	Specifies the time when the packet was sent to the multicast channel for publication. The number represents the number of seconds in UTC time (EPOCH).
SendTimeNS	12	4	Binary	Specifies the number represents the nanosecond portion of UTC time (since EPOCH).

The format of each message within a packet varies according to message type.

However, regardless of the message type, each message will start with a message header consisting of two fields: a two-byte message length, followed by a two-byte message type.

Table 2 Message Header Fields

FIELD	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	-	2	Binary	Size of the message body in bytes including this field.
MsgType	-	2	Binary	Type of message.

1.5.4 Msg Size Field Processing

Customers should not hard code message sizes in feed handlers; instead the feed handler should use the Msg Size field to determine where the next message in the packet begins. This allows the XDP format to accommodate different market needs for data content and allow the format to be more agile.

The variable message size allows the feed handler to insulate client code from any future field additions that may not be desired.

For example, if an original format had the following 16-byte message:

Table 3 Example 1

FIELD	OFFSET	SIZE	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message. 16 bytes
Msg Type	2	2	Binary	Type of message Message '999' – Price message example
SourceTime				Second portion of the ME timestamp.
SourceTimeNS	4	4	Binary	Nanosecond portion of the ME timestamp.
SymbolIndex	8	4	Binary	Numerical representation of the symbol.
Price	12	4	Binary	Price of the order see Price Formats. Use the Price scale from the symbol mapping index.

Look at the Msg Size field to know where the next record will be.

Now the new format adds a new four-byte volume field adjusting the Msg Size to 20 bytes.

The feed handler code automatically is prepared for the 20-byte format without any work, allowing for the receiving application to either continue to read on the first 16 bytes passed to it or develop to read the new field.

Table 4 Example 2

FIELD	OFFSET	SIZE	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message. 20 Bytes
Msg Type	2	2	Binary	Type of message Message '999' – Price message example
SourceTime				Second portion of the ME timestamp.
SourceTimeNS	4	4	Binary	Nanosecond portion of the ME timestamp.
SymbolIndex	8	4	Binary	Numerical representation of the symbol.
Price	12	4	Binary	Price of the order, see Price Formats . Use the Price scale from the symbol mapping index.
Volume	16	4	Binary	Size of the order.

Look at the Msg
Size field to know
where the next
record will be.

1.5.5 Stream ID Message

In all XDP Options multicast channels, each Packet Header is immediately followed by a Stream ID Message. This message's Stream ID field identifies which stream in the channel this packet belongs to.

By combining the Stream ID with the Sequence Number in the Packet Header, the client can uniquely identify each packet and can detect gaps (lost packets) within the stream.

See [Streams and Sequence Numbers](#) for more information.

Note that streams are not applicable to TCP connections to the Request Server, so Stream ID messages are not sent by either the client or by the Request Server.

1.5.6 Sequence Numbers

When reading the multicast channels, client applications should check the Stream ID and Sequence Number of every packet received.

Sequence numbers are unique for each stream, although they do not increase monotonically. Each new sequence number is incremented not by 1, but by the number of messages in the previous packet. So for example if a packet has Stream ID A, Sequence Number 60, and Num Msgs 5, then the next packet to arrive in Stream A will have Sequence Number 65.

You can think of sequence numbers not as packet sequence numbers, but as message sequence numbers. For lower bandwidth, message sequence number fields are not explicitly expressed in each message, but once only in the packet header.

1.5.7 Line Arbitration

Each channel is duplicated and published to two distinct multicast groups for redundancy. The redundant groups are referred to as line A and line B.

Each channel is published in duplicate, Line A and line B are identical in terms of:

- Packet contents
- Sequence numbers

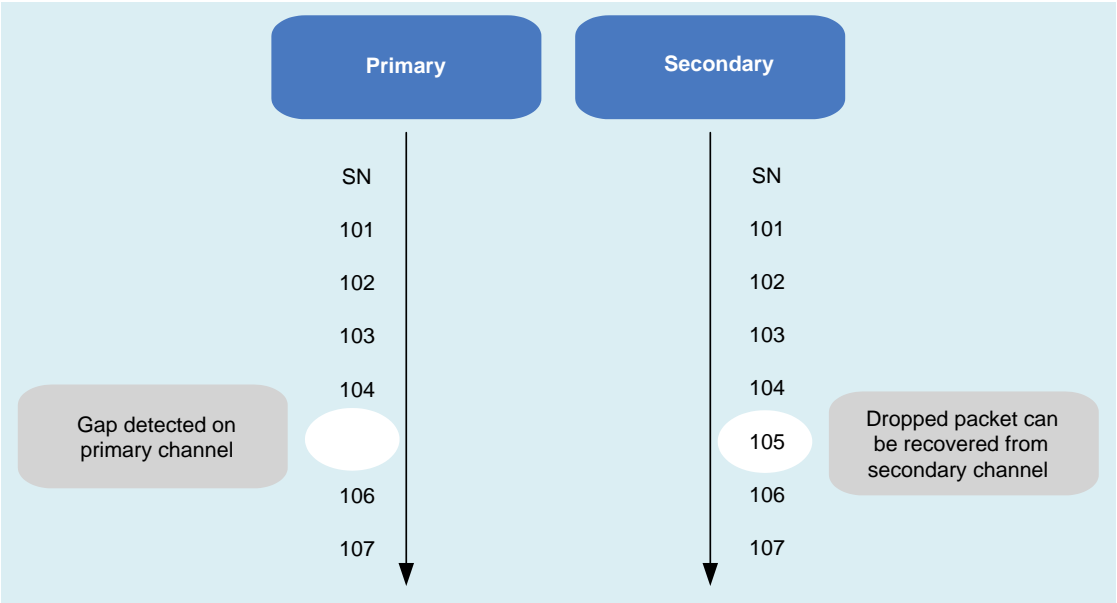
- Order in which the packets are sent

Client applications are advised to read both lines in real-time and process the packets that arrive first, regardless of whether they come from line A or line B. Packets that have lower-than-expected sequence numbers should be discarded.

Using this algorithm, if a packet has been dropped from line A, it can still be read from line B, and vice versa.

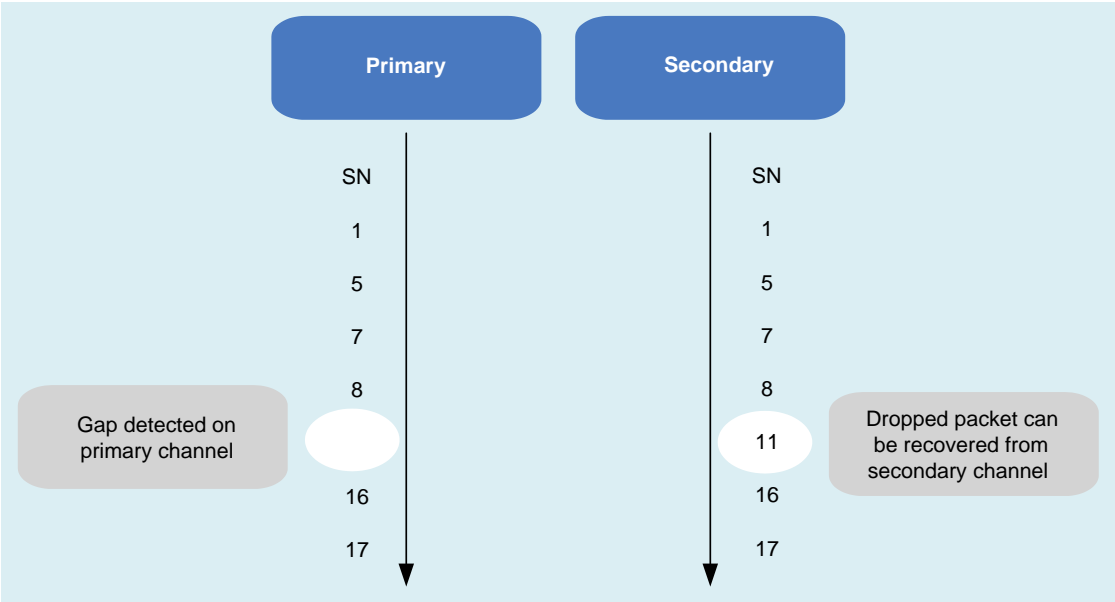
In the rare case where a packet is dropped from both lines, it is necessary to use the refresh mechanism to re-sync the market in the stream that gapped. (see [Refresh Functionality](#))

Figure 2 Packet Header Sequence Numbers (Assuming One Msg per Pkt, all one stream)



Assuming the message sequence below:

FIELD NAME	SEQNUM	NUMMSGs	NEXT EXPECTED SEQNUM
Packet 1	1	4	5
Packet 2	5	2	7
Packet 3	7	1	8
Packet 4	8	3	11
Packet 5	11	1	12
Packet 6	12	4	16
Packet 7	16	1	17



1.5.8 Date and Time Conventions

Dates and times use UTC (Universal Time, Coordinated) EPOCH. For example Wednesday 12/1/09 22:05:17.000 UTC is indicated as 1259791537.

Time stamps are provided over two fields: one that identifies the whole number of seconds in UTC time, and a second that contains the nanosecond offset.

1.5.9 SeriesSeqNum

The SeriesSeqNum field in the message bodies provides a sequence number on an individual series basis and allows users to see if any messages were missed during a session or received out of sequence (audit trail). It also allows users to detect that a message is out of sequence, and ignore that message for their book refresh logic, as appropriate.

1.5.10 Price Formats

All price fields are sent in Binary format.

Prices in this feed are represented by two fields, separating the denominator and the numerator. All prices in the feed share a common denominator (unless otherwise specified), which is represented in the PriceScaleCode.

The PriceScaleCode field value represents the common denominator for the following formula:

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

For example, a price of 27.56 is represented by a Numerator of 2756 and a PriceScaleCode equals to 2.

1.6 HEARTBEATS

1.6.1 Heartbeat Packets in Multicast Feeds

Heartbeat packets are sent over the multicast channels during the startup sequence and in periods of total inactivity (see [System Behavior on Start and Restart](#) for more information). Heartbeat packets are sent per stream, and therefore consist of a packet header followed by a Stream ID message.

- **Heartbeat packet header layout**

FIELD	VALUE
PktSize	16 Bytes
DeliveryFlag	1 (Heartbeat Message Only)
NumberMsgs	1, due to the Stream ID msg (type 455) that follows
SeqNum	Next expected sequence number
SendTime	
SendTimeNS	

1.6.2 TCP Heartbeat Processing Notes

The following applies to TCP connections to the Request Server only.

Clients receive TCP Heartbeat messages (Msg Type 12) periodically (nominally every 60 seconds) whenever they have a TCP/IP connection with the Request Server.

Clients must respond to any heartbeat received by sending back a TCP Heartbeat message within 5 seconds. If no response is received by the Request Server within this timeframe, the server will close the TCP/IP connection.

The Request Server uses this mechanism to close improperly dropped connections, which can arise as a result of client or network hardware failures.

Request Server Heartbeats



1.7 OPERATIONAL INFORMATION

The following measures are in place to safeguard against unexpected system failures.

1.7.1 System Behavior on Start and Restart

At the start of the day, each stream will send the following messages:

- 10 Heartbeats
- Sequence Number Reset message (Msg Type 1), sequence number is set to 1

This means that if channel X is composed of 3 streams, subscribers to channel X will see a total of 30 Heartbeats and 3 Sequence Number Reset messages.

1.7.2 Client System Failure

In case of client system failure, the client should restart, re-subscribe to the multicast channels, and rely on the inline Refresh messages to re-sync with the current state of the market.

For more information, see [Refresh Functionality](#).

2. XDP Messages and Products

This chapter provides details that are specific to each of the market data products, including formats for each message type.

2.1 PRODUCTS

XDP Options messages are grouped into Products. The messages clients receive depend on the products to which they subscribe. Currently, the XDP Options feed offers the following three products.

- Outright Top of Book Product

This product covers outright options symbols and includes the following message types: Outright Quotes, Outright Trades, Outright Trade Corrections, Outright Trade Cancellations, Outright Imbalance, Outright RFQ, Outright EOD Summary, Outright Symbol Status, Underlying Symbol Status, Refresh Outright Quotes, Refresh Outright Trades, Refresh Outright Trade Corrections and Refresh Outright Imbalance.

- Series Depth of Book Product

This product covers outright options symbols and includes the following message types: Outright Depth of book, Outright Symbol Status, Underlying Symbol Status, Refresh Outright Depth of book.

- Complex Top of Book Product

This product covers complex options symbols and includes the following message types: Complex Quotes, Complex Trades, Complex Imbalance, Complex RFQ, Complex EOD Summary, Complex Symbol Status, Underlying Symbol Status, Complex Symbol Definition, Refresh Complex Quotes, Refresh Complex Trades and Refresh Complex Imbalance.

Furthermore, Symbol download messages, including Underlying Index Mapping, Series Index Mapping and Complex Symbol Definition, as well as all Administrative messages are provided with all products.

2.2 MESSAGES

The following section provides an overview of the different message types published by XDP for the NYSE Options markets.

The message types are broken down into five general categories:

- Outright messages
- Complex messages
- Index mapping messages
- Control messages
- Refresh messages

2.2.1 Outright Options Symbol Messages Published

The following XDP messages are published for outright options symbols in real-time, on a push-based publishing model.

- Quotes: Best Buy and Sell price levels in book (quotes and orders). Double-sided message.
- Depth of Book: 3 best Buy and Sell price levels in book (quotes and orders). Each single-sided message provides all three limits (no insert/delete messages are used).
- Trade: all last sales
- Trade Cancel: cancellation of previously published trades
- Trade Correction: correction of previously published trades

- Security Status: symbol events
- Imbalance: theoretical opening price and quantities, published during Pre-opening or symbol Halts
- Crossing RFQ: Request-for-quote/auction mechanism specific to Options
- Summary Message: trading highlights of the day, published once at EOD

2.2.2 Complex Options Symbol Messages Published

The following XDP messages are published for complex symbols in real-time, on a push-based publishing model.

- Quotes message: Best Buy and Sell limits in book (orders only). Double-sided message.
- Trade message: all last sales
- Security Status message: complex symbol events
- Imbalance message: theoretical opening price and quantities, published during Pre-opening and symbol Halts
- Crossing RFQ message: Request-for-quote/auction mechanism specific to Options
- Summary message: trading highlights of the day. Published once at EOD.

2.2.3 Index Mapping Messages

The following index mapping messages are published at the start of every trading day over the multicast channels, when the system comes up. The same messages are also available via a TCP download request.

- Underlying Symbol Index Mapping
- Series Index Mapping
- Complex Leg Definition message

When published intra-day, these messages contain single, newly-created symbols (Outright and Complex). When published as part of the SOD symbol mapping transmission, they provide lists of all active symbols.

2.2.4 Control Messages

The following control messages are available.

- Stream ID message
- TCP login request message
- TCP login response message
- TCP logout request message
- TCP heartbeat message
- TCP heartbeat response message
- Test request message
- Test response message

- Underlying symbol index mapping request message
- Series symbol index mapping request message
- Complex symbol index mapping request message
- Index mapping request response message
- Sequence number reset message
- Message unavailable message

2.2.5 Refresh Messages

The following snapshot Refresh message are published for both Outright and Complex option symbols.

- Refresh outright quotes message
- Refresh outright depth of book message
- Refresh outright trade message
- Refresh outright imbalance message
- Refresh outright trade correction
- Refresh complex quotes message
- Refresh complex trade message
- Refresh complex imbalance message

3. Outright Options Symbol Message Formats

The following section provides formats of messages for outright options symbols.

3.1 OUTRIGHT QUOTE MESSAGE STRUCTURE – MSG TYPE 401

The Quotes message is a dual-sided message providing best bid and offer limits (quotes and orders combined) with aggregated volumes at each limit. The message also indicates aggregated Customer volumes within the overall volumes.

The table below describes the body fields of a Quote message, MsgType 401

This message is sent through the Series Top of Book channel.

Quote Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: ▪ 40 Bytes
MsgType	2	2	Binary	Type of message: ▪ 401 – Outright Quotes Message
SourceTime	4	4	Binary	Seconds portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the outright options symbol.
AskPrice	20	4	Binary Signed	Best Ask price. Should be used with the Price scale from the symbol mapping index.
BidPrice	24	4	Binary Signed	Best Bid price. Should be used with the Price scale from the symbol mapping index
AskShares	28	2	Binary	Total quantity available at the above Ask price.
BidShares	30	2	Binary	Total quantity available at the above Bid price.
AskCustomerShares	32	2	Binary	Total quantity of 'Customer' orders available at the above Ask price.
BidCustomerShares	34	2	Binary	Total quantity of 'Customer' orders available at the above Bid price.
QuoteCondition	36	1	ASCII	1 (Regular Trading) 2 (Rotation) 3 (Trading Halted)
Reserved	37	1	Binary	Filler
Reserved	38	2	Binary	Filler

3.2 OUTRIGHT MARKET DEPTH MESSAGE STRUCTURE, BUY – MSG TYPE 403

The Buy Market Depth message is a single-sided message containing the 3 top Buy limits of orders and quotes on outright symbols with aggregated volumes at each limit. Price Levels 2 and 3 are provided as offsets from Level 1. To determine the second and third price levels, subtract the offsets from the price in Level 1.

There is a separate message for Sell Depth. Each single-sided message provides all three limits (no insert/delete messages are used).

This message is sent through the Outright Depth of Book product.

Table below describes the body fields of a Buy Market Depth message, MsgType 403.

Buy Market Depth message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: ▪ 36 bytes
MsgType	2	2	Binary	Type of message: ▪ 403 – Buy Market Depth
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
First Level	20	4	Binary Signed	Price of First Buy level
Second Level Offset	24	2	Binary	Subtract offset from First level to determine Second Buy level
Third Level Offset	26	2	Binary	Subtract offset from First level to determine Third Buy level
First Volume	28	2	Binary	Total volume at First price level
Second Volume	30	2	Binary	Total volume at Second price level
Third Volume	32	2	Binary	Total volume at Third price level
Reserved	34	2	Binary	Filler

3.3 OUTRIGHT MARKET DEPTH MESSAGE STRUCTURE, SELL – MSG TYPE 405

The Sell Market Depth message is a single-sided message containing the 3 top Sell limits of orders and quotes on outright symbols with aggregated volumes at each limit. Price Levels 2 and 3 are provided as offsets from Level 1. To determine the second and third price levels, add the offsets to the price in Level 1.

There is a separate message for Buy Depth.

This message is sent through the Outright Depth of Book product.

Table below describes the body fields of a Sell Market Depth message, MsgType 405.

Sell Market Depth message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: ▪ 36 bytes
MsgType	2	2	Binary	Type of message: ▪ 405 – Sell Market Depth
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
FirstLevel	20	4	Binary Signed	Price of First Sell level
Second Level Offset	24	2	Binary	Add offset to First level to determine Second Sell level
Third Level Offset	26	2	Binary	Add offset to First level to determine Third Sell level
First Volume	28	2	Binary	Total volume at First price level
Second Volume	30	2	Binary	Total volume at Second price level
Third Volume	32	2	Binary	Total volume at Third price level
Reserved	34	2	Binary	Filler

3.4 OUTRIGHT TRADE MESSAGE STRUCTURE – MSG 407

The Outright Trade message is used to publish all Last Sales for outright symbols. This message is sent through the Outright Top of Book product.

The table below describes the body fields of a Trade message, MsgType 407.

Trade Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: ▪ 32 bytes
MsgType	2	2	Binary	Type of message: ▪ 407 – Trade Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
TradeID	20	4	Binary	Unique Trade execution ID.
Price	24	4	Binary Signed	Price of the trade. Use the Price scale from the symbol mapping index.
Volume	28	2	Binary	Volume of the trade in number of contracts
TradeCond1	30	1	ASCII	Blank (regular trade) I (Late report) R (Floor trade) S (ISO sweep trade)
TradeCond2	31	1	ASCII	Complex indicator: P (Complex trade with equity trade) L (Complex trade)

3.5 OUTRIGHT TRADE CANCEL MESSAGE STRUCTURE – MSG TYPE 409

The Trade Cancel message is used to cancel or bust a trade. This message is sent through the Outright Top of Book product. The table below describes the body fields of an Outright Trade Cancel message, MsgType 409.

Trade Cancel or Bust Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 24 bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> 409 – Outright Trade Cancel Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
OriginalTradeID	20	4	Binary	Original TradeID of the trade.

3.6 OUTRIGHT TRADE CORRECTION MESSAGE STRUCTURE – MSG TYPE 411

The Trade Correction message is used to correct a trade. This message is sent through the Outright Top of Book product. The table below describes the body fields of an Outright Trade Correction message, MsgType 411.

Trade Correction Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: ▪ 36 bytes
Msg Type	2	2	Binary	Type of message: ▪ 411 – Outright Trade Correction Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
OriginalTradeID	20	4	Binary	Original TradeID of the trade.
TradeID	24	4	Binary	New unique TradeID used to identify the corrected trade.
Price	28	4	Binary Signed	Price of the order. Use the Price scale from the symbol mapping index.
Volume	32	2	Binary	volume of the trade in number of contracts.
TradeCond1	34	1	ASCII	Blank (regular trade) I (Late report) R (Floor trade) S (ISO sweep trade)
TradeCond2	35	1	ASCII	Complex indicator P (Complex trade with equity trade) L (Complex trade)

3.7 OUTRIGHT IMBALANCE MESSAGE – MSG TYPE 413

The Outright Imbalance message is sent during the Pre-opening phase or a symbol Halt to provide indicative opening price and net imbalance of all orders at that price which are orders eligible for next Auction. It also indicates the type of Auction to follow. Imbalance values are calculated using both orders and quotes.

This message is sent through the Outright Top of Book product.

The table below describes the body fields of an Outright Imbalance Message, MsgType 413.

Outright Imbalance Message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: ▪ 36 bytes
Msg Type	2	2	Binary	Type of message. ▪ 413 – Outright Imbalance
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
ReferencePrice	20	4	Binary Signed	Indicative opening price.
PairedQty	24	2	Binary	Paired off quantity at the indicative opening price (Indicative matching quantity)
TotalImbalanceQty	26	2	Binary	Total imbalance quantity at the indicative opening price.
MarketImbalanceQty	28	2	Binary	Total market order imbalance at the indicative opening price.
AuctionTime	30	2	Binary	Projected Auction Time (hhmm)
AuctionType	32	1	ASCII	O (Opening) M (Market) H (Halt)
ImbalanceSide	33	1	ASCII	Side of the imbalance Buy/sell. Valid Values: B (Buy) S (Sell) Blank (No imbalance)
Reserved	34	2	Binary	Filler

3.8 OUTRIGHT CROSSING RFQ MESSAGE – MSG 415

The Outright Crossing RFQ (Request for Quote) message is sent out in the event of an auction on an outright order.

This message is sent through the Outright Top of Book product.

The table below describes the body fields of an RFQ Quote message, MsgType 415.

Crossing RFQ message fields

Field	Offset	Size (Bytes)	Format	Description
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 28 bytes
MsgType	2	2	Binary	Type of message. <ul style="list-style-type: none"> 415 - Outright Crossing RFQ
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceNS	8	4	Binary	Second portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
Side	20	1	ASCII	Side of the RFQ B (Buy) S (Sell)
Reserved	21	1	Binary	Filler
Shares	22	2	Binary	Total quantity
Price	24	4	Binary Signed	Price of crossing transaction

3.9 OUTRIGHT SUMMARY MESSAGE STRUCTURE – MSG 417

The Outright summary message provides trading highlights of the day. It is sent out once at the end of the trading day as soon as a symbol closes.

Note: If no quote or last sale occurred on a symbol, no End of Day Summary is generated.

This message is sent through the Outright Top of Book product.

The table below describes the body fields of an Outright Summary message, MsgType 417.

Outright Summary Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 40 bytes
Msg Type	2	2	Binary	Type of message. <ul style="list-style-type: none"> 417 – Outright Summary message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTime NS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
HighPrice	20	4	Binary Signed	Exchange high price of the symbol for the day. Use the Price scale from the symbol mapping index.
LowPrice	24	4	Binary Signed	Exchange Low price of the symbol for the day. Use the Price scale from the symbol mapping index.
Open	28	4	Binary	Exchange Opening price of the symbol for the day. Use the Price scale from the symbol mapping index.
Close	32	4	Binary Signed	Exchange Closing price of the symbol for the day. Use the Price scale from the symbol mapping index.
TotalVolume	36	2	Binary	Exchange cumulative volume for the symbol throughout the day.
Reserved	38	2	Binary	Filler

3.10 UNDERLYING STATUS MESSAGE STRUCTURE – MSG 419

The Underlying Status message is used to inform the subscribers of changes in the status of an Underlying symbol.

This message is sent through both the Outright Top of Book and the Complex Top of Book products.

The table below describes the body fields of a Security Status message, msg type 419.

Underlying Status message fields

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Message size: ■ 24 bytes
MsgType	2	2	Binary	Type of message. ■ 419 – Underlying Status Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
UnderlyingIndex	12	4	Binary	Numerical representation of the underlying symbol.
UnderlyingSeqNum	16	4	Binary	Sequence number of messages for the underlying symbol.
SecurityStatus	20	1	ASCII	S Halt U Unhalt O Open indication X Close indication
Halt Condition	21	1	ASCII	Not applicable
Reserved	22	2	Binary	Filler

3.11 OUTRIGHT SERIES STATUS MESSAGE STRUCTURE – MSG 421

The Series Status message is used to inform the subscribers of changes in symbol status of an outright symbol.

This message is sent through the Outright Top of Book channel.

The table below describes the body fields of a Series Status message, msg type 421.

Series Status message fields

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 24 bytes
MsgType	2	2	Binary	Type of message. <ul style="list-style-type: none"> 421 – Security Status Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the Outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
SecurityStatus	20	1	ASCII	L light up a dark N Open indication (dark) O Open X Close S Halt U Unhalt T Unhalt dark P Beginning of RFQ auction Q End of RFQ auction
HaltCondition	21	1	ASCII	Not applicable
Reserved	22	2	Binary	Filler

4. Complex Message Formats

The following section provides formats of messages for complex symbols.

4.1 COMPLEX QUOTE MESSAGE STRUCTURE – MSG TYPE 423

The Complex Quotes message is a dual-sided message providing best bid and offer limits of orders on complex symbols with aggregated volumes at each limit. The message also indicates aggregated Customer volumes within the overall volumes.

The table below describes the body fields of a Complex Quote message, MsgType 423.

This message is sent through the Complex Top of Book channel.

Quote Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 40 Bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> 423 – Complex Quote
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
ComplexIndex	12	4	Binary	Numerical representation of the complex symbol.
SymbolSeqNum	16	4	Binary	Sequence number of messages for the complex symbol.
AskPrice	20	4	Binary Signed	Best Ask price. Should be used with the Price scale from the symbol mapping index.
BidPrice	24	4	Binary Signed	Best Bid price. Should be used with the Price scale from the symbol mapping index.
AskShares	28	2	Binary	total quantity available at the above Ask price.
BidShares	30	2	Binary	total quantity available at the above Bid price.
AskCustomerShares	32	2	Binary	total quantity of 'Customer' orders available at the above Ask price.
BidCustomerShares	34	2	Binary	total quantity of 'Customer' orders available at the above Bid price.
QuoteCondition	36	1	ASCII	1 (Regular Trading) 2 (Rotation) 3 (Trading Halted)
Reserved1	37	1	Binary	Filler
Reserved2	38	2	Binary	Filler

4.2 COMPLEX TRADE MESSAGE STRUCTURE – MSG 425

The Complex Trade message is used to publish all complex Last Sales. This message is sent through the Complex Top of Book product.

The table below describes the body fields of a Trade message, MsgType 425.

Trade Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: ▪ 32 bytes
MsgType	2	2	Binary	Type of message: ▪ 425 – Trade Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
ComplexIndex	12	4	Binary	Numerical representation of the complex symbol.
SymbolSeqNum	16	4	Binary	Sequence number of messages for the complex symbol.
TradeID	20	4	Binary	(blank) n/a for complex symbols
Price	24	4	Binary Signed	Price of the trade. Can be negative. Use the Price scale from the symbol mapping index.
Volume	28	2	Binary	Volume of the trade in actual number of contracts
TradeCond1	30	1	ASCII	Blank (regular trade) I (Late report) R (Floor trade) S (ISO sweep trade)
TradeCond2	31	1	ASCII	(blank) n/a for complex symbols

4.3 COMPLEX IMBALANCE MESSAGE – MSG TYPE 427

The Complex Imbalance message is sent during the Pre-opening phase or a symbol Halt to provide indicative opening price and net imbalance of all orders at that price which are orders eligible for next Auction. It also indicates the type of Auction to follow.

This message is sent through the Complex Top of Book product.

The table below describes the body fields of a Complex Imbalance Message, MsgType 427.

Imbalance Message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: ▪ 36 bytes
Msg Type	2	2	Binary	Type of message. ▪ 427 – Complex Imbalance
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
ComplexIndex	12	4	Binary	Numerical representation of the complex symbol.
SymbolSeqNum	16	4	Binary	Sequence number of messages for the complex symbol.
ReferencePrice	20	4	Binary Signed	Indicative opening price
PairedQty	24	2	Binary	paired off quantity at the indicative opening price (Indicative matching quantity).
TotalImbalanceQty	26	2	Binary	total imbalance quantity at the indicative opening price.
MarketImbalanceQty	28	2	Binary	Total market order imbalance at the indicative opening price.
AuctionTime	30	2	Binary	Projected Auction Time (hhmm)
AuctionType	32	1	ASCII	O (Opening) M (Market) H (Halt)
ImbalanceSide	33	1	ASCII	Side of the imbalance Buy/sell. Valid Values: B (Buy) S (Sell) Blank (No imbalance)
Reserved	34	2	Binary	Filler

4.1 COMPLEX CROSSING RFQ MESSAGE– MSG 429

The Complex Crossing RFQ (Request for Quote) message is sent out in the event of an auction on a complex order.

This message is sent through the Complex Top of Book product.

The table below describes the body fields of an RFQ Quote message, MsgType 429.

Crossing RFQ message fields

Field	Offset	Size (Bytes)	Format	Description
MsgSize	0	2	Binary	Size of the message body in bytes. ▪ 28 bytes
MsgType	2	2	Binary	Type of message. ▪ 429 – Complex Crossing
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceNS	8	4	Binary	Second portion of the ME timestamp.
ComplexIndex	12	4	Binary	Numerical representation of the complex symbol.
SymbolSeqNum	16	4	Binary	Sequence number of messages for the complex symbol.
Side	20	1	ASCII	Side of the RFQ B (Buy) S (Sell)
Reserved	21	1	Binary	Filler
Shares	22	2	Binary	Total quantity
Price	24	4	Binary Signed	Price of crossing transaction

4.2 COMPLEX SUMMARY MESSAGE STRUCTURE – MSG 431

The Complex summary message provides trading highlights of the day. It is sent out once at the end of the trading day as soon as a symbol closes.

Note: If no quote or last sale occurred for a complex symbol, no End of Day Summary is generated.

This message is sent through the Complex Top of Book channel.

The table below describes the body fields of a Complex Summary message, MsgType 431.

Table 5 Stock Summary Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 40
Msg Type	2	2	Binary	Type of message. <ul style="list-style-type: none"> 431 – Complex Summary Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
ComplexIndex	12	4	Binary	Numerical representation of the complex symbol.
SymbolSeqNum	16	4	Binary	Sequence number of messages for the complex symbol.
HighPrice	20	4	Binary Signed	Exchange high price of the symbol for the day. Use the Price scale from the symbol mapping index.
LowPrice	24	4	Binary Signed	Exchange Low price of the symbol for the day. Use the Price scale from the symbol mapping index.
OpenPrice	28	4	Binary Signed	Exchange Opening price of the symbol for the day. Use the Price scale from the symbol mapping index.
ClosePrice	32	4	Binary Signed	Exchange Closing price of the symbol for the day. Use the Price scale from the symbol mapping index.
TotalVolume	36	2	Binary	Exchange cumulative volume for the symbol throughout the day.
Reserved	38	2	Binary	Filler

4.3 COMPLEX STATUS MESSAGE – MSG 433

The Complex Status message is used to inform the subscribers of changes in symbol status of a complex symbol.

This message is sent through the Complex Top of Book product.

Note: the Underlying Status message is also sent through the Complex Top of Book product.

The table below describes the body fields of a Complex Status message, msg type 433.

Complex Status message fields

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Message size: <ul style="list-style-type: none"> 24 bytes
MsgType	2	2	Binary	Type of message. <ul style="list-style-type: none"> 433 – Complex Status Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
ComplexIndex	12	4	Binary	Numerical representation of the complex symbol.
SymbolSeqNum	16	4	Binary	Sequence number of messages for the complex symbol.
SecurityStatus	20	1	ASCII	L Light up a dark N Open indication (dark) O Open X Close S Halt U Unhalt T Unhalt dark P Beginning of RFQ auction Q End of RFQ auction
Halt Condition	21	1	ASCII	Not applicable
Reserved	22	2	Binary	Filler

5. Index Mapping Message Formats

5.1 UNDERLYING INDEX MAPPING MESSAGE – MSG 435

The Underling Index Mapping message is the first message type sent down every multicast feed at start of day. Each multicast channel sends an Underlying Index Mapping message for every underlying symbol that appears on that channel. If individual series for the same underlying symbol appear in multiple streams, the Underlying Index Mapping message is sent down all relevant channels.

This message is also available via the symbol download request.

The table below describes the body fields of a Symbol Index Mapping message, msg type 435.

Underlying Index Mapping message fields

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: 28 bytes
Msg Type	2	2	Binary	Type of message 435 – Underlying Index Mapping
UnderlyingIndex	4	4	Binary	Numerical representation of the Underlying symbol.
UnderlyingSymbol	8	11	ASCII	Full symbol in NYSE Symbology.
ChannelID	19	1	Binary	Multicast channel ID of the symbols being provided.
Market ID	20	2	Binary	Identifies originating market: 1 (NYSE Cash) 2 (Europe Cash) 3 (NYSE Arca Cash) 4 (NYSE/Arca Options) 5 (NYSE/Arca Bonds) 6 (ArcaEdge) 7 (LIFFE) 8 (NYSE Amex Options) 9 (NYSE MKT Cash)
System ID	22	1	Binary	ID of the Originating System
Exchange Code	23	1	ASCII	Exchanges where it is listed: N (NYSE) P (NYSE Arca) Q (NASDAQ) A (NYSE MKT)

PriceScaleCode	24	1	Binary	Price scale for price conversion of the symbol. See Price Formats.
SecurityType	25	1	ASCII	Type of Security: A (ADR) C (COMMON STOCK) D (DEBENTURES) E (ETF) F (FOREIGN) I (UNITS) M (MISC/LIQUID TRUST) P (PREFERRED STOCK) R (RIGHTS) S (SHARES OF BENEFICIARY INTEREST) T (TEST) U (UNITS) W (WARRANT)
Price Resolution	26	1	Binary	0 (All Penny) 1 (Penny/Nickel) 5 (Nickel/Dime)
Reserved	27	1	Binary	Filler

5.2 SERIES INDEX MAPPING MESSAGE– MSG 437

Series symbols are used to identify options instruments. They are unique to a given Trading unit but not across the entire system. The Series ID must therefore be concatenated with Market ID and system ID to uniquely identify a given instrument.

The Series Index Mapping messages immediately follow the Underlying Index Mapping messages. Each multicast channel sends all the symbols on its channel.

This message is also sent intraday whenever a new outright symbol is created (or was erroneously omitted from the initial spin), and is available via the symbol download request.

The table below describes the body fields of a Series Index Mapping message, msg type 437.

Series Index message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Message size: <ul style="list-style-type: none"> 60 bytes
MsgType	2	2	Binary	Type of message. <ul style="list-style-type: none"> 437 - Series Index Mapping
SeriesIndex	4	4	Binary	Numerical representation of the outright options symbol.
ChannelID	8	1	Binary	multicast channel ID of the symbols being provided.
Reserved	9	1	Binary	Filler
MarketID	10	2	Binary	Identifies originating market: <ul style="list-style-type: none"> 1 (NYSE Cash) 2 (Europe Cash) 3 (NYSE Arca Cash) 4 (NYSE/Arca Options) 5 (NYSE/Arca Bonds) 6 (Global OTC) 7 (LIFFE) 8 (NYSE Amex Options) 9 (NYSE MKT Cash)
SystemID	12	1	Binary	Identifies Trading unit (TU)
Reserved	13	1	Binary	Filler
StreamID	14	2	Binary	Identifies Stream on which this symbol will be updated
UnderlyingIndex	16	4	Binary	Underlying Stock Mapping Index
ContractMultiplier	20	2	Binary	Contract quantity
MaturityDate	22	6	ASCII	YY MM DD
PutOrCall	28	1	Binary	0 (Put) 1 (Call)

XDP Options Client Specification v1.0g

StrikePrice	29	10	ASCII	
PriceScale	39	1	Binary	Decimal places on price
UnderlyingSymbol	40	11	ASCII	full symbol in NYSE symbology of the underlying symbol.
OptionSymbolRoot	51	5	ASCII	OCC root of option symbol
GroupID	56	4	Binary	Group to which the symbol belongs

5.3 COMPLEX SYMBOL DEFINITION MESSAGE – MSG 439

The Complex Symbol Definition message provides a description of the legs in a Complex symbol. The message references a unique Complex symbol ID. A leg can be an option or an equity; the leg type is indicated in the LegSecurityType field. The number of leg definitions is provided in the message. The order of legs is deterministic. The length of the message is variable depending on the number of legs in the complex symbol.

This message is sent as part of the SOD symbol transmission. Each multicast channel sends all the symbols on its channel. This message is also sent intraday whenever a new complex symbol is created, and is available via a symbol download request.

The table below describes the body fields of a Complex Leg Definition message, MsgType 439.

Complex Leg Definition Message Fields

FIELD NAME	OFFS ET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> Variable, 80 max
MsgType	2	2	Binary	Type of message. <ul style="list-style-type: none"> 439 – Complex Symbol Definition
ComplexIndex	4	4	Binary	Numerical representation of the complex symbol.
ComplexSymbol	8	21	ASCII	Complex Symbol. This is an alpha numeric symbol which can be used for a unique strategy until one of the legs expire.
ChannelID	29	1	Binary	multicast channel ID of the symbols being provided.
MarketID	30	2	Binary	Identifies originating market: 1 (NYSE Cash) 2 (Europe Cash) 3 (NYSE Arca Cash) 4 (NYSE/Arca Options) 5 (NYSE/Arca Bonds) 6 (ArcaEdge) 7 (LIFFE) 8 (NYSE Amex Options) 9 (NYSE MKT Cash)
SystemID	32	1	Binary	Identifies Trading unit (TU)
Reserved	33	1	Binary	Filler
StreamID	34	2	Binary	Identifies Stream on which this symbol will be updated
NoOfLegs	36	2	Binary	Number of legs in complex symbol

				1 – 5
Reserved	38	2	Binary	Filler
Leg definition – max of 5				
SymbolIndex	40	4	Binary	Series index if Security type is Option Underlying index if Security type is Equity
LegRatioQty	44	2	Binary	Leg ratio
Side	46	1	ASCII	Leg side B (Buy) S (Sell)
SecurityType	47	1	ASCII	Leg Security Type O (Options Series leg) E (Equity stock leg)

6. Control Message Formats

The following section provides the format of Control messages. Control messages allow conversing parties to exchange session-specific information, such as “Reset sequence number”.

6.1 STREAM ID MESSAGE – MSG 455

In all multicast channels, a Stream ID Message is sent immediately after each packet header to uniquely identify the source or Stream ID of the data in the packet. Stream ID Messages are not applicable in TCP request/response connections, and so are not used by either the client or the Request Server.

The table below describes the body fields of a Stream ID message, msg type 455.

Stream ID message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 8 Bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> 455 – Stream ID Header Message
StreamID	4	2	Binary	Represents the Stream ID for this packet update. The Stream ID represents a unique source (specific thread within a publisher task).
Reserved	6	2	Binary	Filler

6.2 TCP LOGIN REQUEST MESSAGE – MSG 456

This message is sent by clients to log into the Request Server. The server responds with a Login Request Response message, msg type 457.

The table below describes the body fields of a TCP Login Request message, msg type 456.

TCP Login Request message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 28 Bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> 456 – Login Request Message
SourceID	4	10	ASCII	Name of the source requesting login. This field is up to 9 characters, null terminated.
Password	14	12	ASCII	Password of the requestor
Reserved	26	2	Binary	Filler

6.3 TCP LOGIN REQUEST RESPONSE MESSAGE – MSG 457

This message is sent in response to a client request to log into the Refresh Server. It indicates whether or not the login request has been accepted. In the event the request is rejected, the message indicates the reason for the rejection. A request may be rejected for the following reasons:

- The requestor is not authorized or has an invalid password
- The server has exceeded the maximum connection limit for this port
- The connection has timed out (client connects and does not log in within 30 seconds).

In the event the request is rejected, the Refresh Server closes the socket connection after sending this message.

The table below describes the body fields of a TCP Login Request Response message, msg type 457.

TCP Login Request Response message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> ▪ 8 Bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> ▪ 457 – Login Request Response Message
ResponseCode	4	1	ASCII	A (Accepted) B (Rejected)
RejectCode	5	1	ASCII	(blank) A (Not authorized) M (Maximum server connections reached) T (Timeout)
Reserved	6	2	Binary	Filler

6.4 TCP HEARTBEAT MESSAGE - MSG 12

This message is sent by the Request Server to the client as soon as a TCP connection is established, and is republished approximately every 60 seconds to indicate the session is active.

A responding TCP Heartbeat message from the client system must be received by the server within 5 seconds of sending the heartbeat, or else the session will be terminated.

The table below describes the body fields of a TCP Heartbeat message, msg type 12.

TCP Heartbeat message fields

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> ▪ 16 Bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> ▪ 12 – Heartbeat
SourceID	4	10	ASCII	Name of the connected client. This field is up to 9 characters, null terminated.
Reserved	14	2	Binary	

6.5 TEST REQUEST MESSAGE – MSG 458

This message can be sent by clients to request a response from the Request Server during periods of inactivity. The client can specify a text message for the server to echo back its response.

The table below describes the body fields of a Test Request message, msg type 458.

Test Request Message Format

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 12 Bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> 458 – Test Request Message
TestMessage	4	8	ASCII	Text to be echoed back

6.6 TEST REQUEST RESPONSE MESSAGE – MSG 459

This message is sent back from the Request Server to a client in response to a Test Request message.

The table below describes the body fields of a Test Request Response message, msg type 459.

Test Request Message Format

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 12 Bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> 459 – Test Request Response Message
TestMessage	4	8	ASCII	Client text to echo from the Test Request message

6.7 UNDERLYING INDEX MAPPING REQUEST MESSAGE (MSG TYPE 441)

This message is sent by Subscribers via TCP/IP to request Underlying index mapping messages.

If the UnderlyingIndex field in the message is populated with a specific value, the response will contain the Underlying index requested. In this case, any value in the ChannelID field is ignored.

If the UnderlyingIndex field in the message is populated with the value '0', the response will contain all the Underlying indices on the channel defined in the ChannelID field.

If the ChannelID field is populated with the value '0', the request will contain all the Underlying symbols on the Exchange. In this case, any value in the UnderlyingIndex field is ignored.

The Request Server responds to this message with a Request Response Message, to indicate the outcome of the request. If the request is accepted, the Request Response Message is followed by the index mapping messages requested. When the download is complete, a second Request Response Message is sent to indicate that the transmission is finished.

The table below describes the body fields of a Underlying Index Mapping Request message, msg type 441.

Underlying Index Mapping Request message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 20 Bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> 441 – Underlying Index Mapping Request Message
UnderlyingIndex	4	4	Binary	Numerical representation of the underlying symbol being requested. Alternatively, it can be set to zero in order to request all underlying symbols in the channel (multicast group).
SourceID	8	10	ASCII	Name of the requestor.
ChannelID	18	1	Binary	multicast channel ID of the symbols being requested. Alternatively, it can be set to zero in order to request all underlying symbols on the Exchange.
Reserved	19	1	Binary	Filler

6.8 SERIES INDEX MAPPING REQUEST MESSAGE (MSG TYPE 443)

This message is sent by Subscribers via TCP/IP to request Series index mapping messages.

If the SeriesIndex field in the message is populated with a specific value, the response will contain the Series index requested. In this case, any value in the ChannelID field is ignored.

If the SeriesIndex field in the message is populated with the value '0', the response will contain all the Series indices on the channel defined in the ChannelID field.

If the ChannelID field is populated with the value '0', the request will contain all the Outright options symbol on the Exchange. In this case, any value in the SeriesIndex field is ignored.

The Request Server responds to this message with a Request Response Message, to indicate the outcome of the request. If the request is accepted, the Request Response Message is followed by the index mapping messages requested. When the download is complete, a second Request Response Message is sent to indicate that the transmission is finished.

The table below describes the body fields of a Series Index Mapping Request message, msg type 443.

Series Index Mapping Request message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 20 bytes
MsgType	2	2	Binary	Type of message. <ul style="list-style-type: none"> 443 – Series Index Mapping Request Message
SeriesIndex	4	4	Binary	Numerical representation of the outright options symbol being requested. Alternatively, it can be set to zero in order to request all outright options symbol in the channel (multicast group).
SourceID	8	10	ASCII	Name of the requestor.
ChannelID	18	1	Binary	multicast channel ID where you requesting the index map from.
Reserved	19	1	Binary	Filler

6.9 COMPLEX SYMBOL DEFINITION REQUEST MESSAGE (MSG TYPE 445)

This message is sent by Subscribers via TCP/IP to request Complex symbol definition messages.

If the ComplexSymbolIndex field in the message is populated with a specific value, the response will contain the Complex symbol definition index requested. In this case, any value in the ChannelID field is ignored.

If the ComplexSymbolIndex field in the message is populated with the value '0', the response will contain all the Series indices on the channel defined in the ChannelID field.

If the ChannelID field is populated with the value '0', the request will contain all the Complex Symbols on the Exchange. In this case, any value in the ComplexSymbolIndex field is ignored.

The Request Server responds to this message with a Request Response Message, to indicate the outcome of the request. If the request is accepted, the Request Response Message is followed by the index mapping messages requested. When the download is complete, a second Request Response Message is sent to indicate that the transmission is finished.

The table below describes the body fields of a Underlying Index Mapping Request message, msg type 445.

Complex Index Mapping Request message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 20 Bytes
MsgType	2	2	Binary	Type of message. <ul style="list-style-type: none"> 445 – Complex Index Mapping Request Message
ComplexSymbolIndex	4	4	Binary	Numerical representation of the complex symbol being requested. Alternatively, it can be set to zero in order to request all complex symbols in the channel (multicast group).
SourceID	8	10	ASCII	Name of the requestor.
ChannelID	18	1	Binary	Multicast channel ID of the complex symbols being requested. Alternatively, it can be set to zero in order to request all complex symbols on the Exchange.
Reserved	19	1	Binary	Filler

6.10 REQUEST RESPONSE MESSAGE (MSG TYPE 447)

This message will be sent via TCP/IP in response to the client's request for symbol mapping messages.

If the request is accepted and the mapping messages provided, this message is sent again immediately at the end of the transmission to indicate that the download is complete.

If the request is rejected, the reason for the rejection is indicated in this message.

The table below describes the body fields of a Request Response message, msg type 447.

Request Response Message Fields

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Message size: <ul style="list-style-type: none"> 16 bytes
MsgType	2	2	Binary	Type of message. <ul style="list-style-type: none"> 447 –Request Response Message
SourceID	4	10	ASCII	Name of the source requesting retransmission.
ChannelID	14	1	Binary	multicast channel ID
Status	15	1	Binary	Outcome of the request. Valid values: <ul style="list-style-type: none"> 0 – Message was accepted 1 – Rejected due to an Invalid Source ID 4 – Rejected due to maximum number of refresh requests in a day 6 – Rejected due to an Invalid Channel ID 8 – Underlying download complete 9 – Series download complete 10 – Complex download complete

6.11 TCP LOGOUT REQUEST MESSAGE - 460

The TCP Logout Request message can be sent by the client to the Request Server as an optional means of disconnecting. The Request Server responds by disconnecting the client.

The other alternative is for the client to simply disconnect when all needed information is received.

Note that if the client fails to respond to any TCP Heartbeat message within five seconds, the server will close the socket.

The table below describes the body fields of a TCP Logout Request message, msg type 460.

TCP Logout Request Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 4 bytes
MsgType	2	2	Binary	Type of message. <ul style="list-style-type: none"> 460 – Logout Request

6.12 SEQUENCE NUMBER RESET (MSG TYPE 1)

This message is sent to 'reset' the Packet Sequence Number at start of day or following to a failover. Note that this message will contain a valid sequence number.

The table below describes the body fields of a Sequence Number Reset message, Msg Type 1.

Sequence Number Reset message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 16 Bytes
MsgType	2	2	Binary	Type of message. <ul style="list-style-type: none"> 1 – Sequence Number Reset
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
ProductID	12	1	Binary	The product ID used in the XDP header to identify the NYSE Euronext feed.
ChannelID	13	1	Binary	multicast channel ID over which the packet was sent.
Reserved	14	2	Binary	Filler

7. Refresh Message Formats

7.1 REFRESH OUTRIGHT QUOTE MESSAGE STRUCTURE – MSG TYPE 501

The Refresh Quotes message is sent in the event no quote messages are published for the symbol for 2 consecutive minutes. It has the same field content as the last published Outright Quote message, with the following exceptions:

- MsgType is 501
- SeriesSeqNum increments on every publication

This message is sent through the Outright Top of Book channel.

Refresh Quote Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 40 bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> 501 – Refresh Outright Quotes message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
AskPrice	20	4	Binary Signed	Best Ask price. Should be used with the Price scale from the symbol mapping index.
BidPrice	24	4	Binary Signed	Best Bid price. Should be used with the Price scale from the symbol mapping index
AskShares	28	2	Binary	total quantity available at the above Ask price.
BidShares	30	2	Binary	total quantity available at the above Bid price.
AskCustomerShares	32	2	Binary	total quantity of 'Customer' orders available at the above Ask price.
BidCustomerShares	34	2	Binary	total quantity of 'Customer' orders available at the above Bid price.
QuoteCondition	36	1	ASCII	1 (Regular Trading) 2 (Rotation) 3 (Trading Halted)
Reserved	37	1	Binary	Filler
Reserved	37	2	Binary	Filler

7.2 REFRESH OUTRIGHT MARKET DEPTH MESSAGE STRUCTURE, BUY – MSG TYPE 503

The Refresh Market Depth Buy message is sent in the event no Buy Market Depth messages are published for the symbol for 2 consecutive minutes. It has the same field content as the last published Outright Quote message, with the following exceptions:

- MsgType is 503
- SeriesSeqNum increments on every publication

This message is sent through the Outright Depth of Book product.

Refresh Buy Market Depth message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 36 bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> 503 – Refresh Buy Market Depth
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
First Level	20	4	Binary Signed	Price of First Buy level
Second Level Offset	24	2	Binary	Price offset from First level to determine Second Buy level
Third Level Offset	26	2	Binary	Price offset from First level to determine Third Buy level
First Volume	28	2	Binary	Total volume at First price level
Second Volume	30	2	Binary	Total volume at Second price level
Third Volume	32	2	Binary	Total volume at Third price level
Reserved	34	2	Binary	Filler

7.3 REFRESH OUTRIGHT MARKET DEPTH MESSAGE STRUCTURE, SELL – MSG TYPE 505

The Refresh Market Depth Sell message is sent in the event no Sell Market Depth messages are published for the symbol for 2 consecutive minutes. It has the same field content as the last published Sell Market Depth message, with the following exceptions:

- MsgType is 505
- SeriesSeqNum increments on every publication

This message is sent through the Outright Depth of Book product.

Refresh Sell Market Depth message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: ▪ 36 bytes
MsgType	2	2	Binary	Type of message: ▪ 505 – Refresh Sell Market Depth
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
First Level	20	4	Binary Signed	Price of First Buy level
Second Level Offset	24	2	Binary	Price offset from First level to determine Second Sell level
Third Level Offset	26	2	Binary	Price offset from First level to determine Third Sell level
First Volume	28	2	Binary	Total volume at First price level
Second Volume	30	2	Binary	Total volume at Second price level
Third Volume	32	2	Binary	Total volume at Third price level
Reserved	34	2	Binary	Filler

7.4 REFRESH OUTRIGHT TRADE MESSAGE STRUCTURE – MSG 507

The Refresh Outright Trade message is sent in the event no Outright Trade messages are published for the symbol for 2 consecutive minutes. It has the same field content as the last published Outright Trade message (assuming no intervening corrections), with the following exceptions:

- MsgType is 507
- SeriesSeqNum increments on every publication

This message is sent through the Outright Top of Book product.

Refresh Outright Trade Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: ▪ 32 bytes
MsgType	2	2	Binary	Type of message: ▪ 507 – Refresh Outright Trade Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the outright options symbol.
TradeID	20	4	Binary	The TradeID identifies a unique Trade execution.
Price	24	4	Binary Signed	Price of the trade. Use the Price scale from the symbol mapping index.
Volume	28	2	Binary	Volume of the trade in number of contracts
TradeCond1	30	1	ASCII	Blank (regular trade) I (Late report) R (Floor trade) S (ISO sweep trade)
TradeCond2	31	1	ASCII	Complex indicator: P (Complex trade with equity trade) L (Complex trade)

7.5 REFRESH OUTRIGHT IMBALANCE MESSAGE – MSG TYPE 509

The Refresh Outright Imbalance message is sent in the event no Outright Imbalance messages are published for the symbol for 2 consecutive minutes during Pre-opening or a symbol Halt. It has the same field content as the last published Outright Imbalance message, with the following exceptions:

- MsgType is 509
- SeriesSeqNum increments on every publication

This message is sent through the Outright Top of Book product.

Refresh Outright Imbalance Message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: <ul style="list-style-type: none"> ▪ 36 bytes
Msg Type	2	2	Binary	Type of message. <ul style="list-style-type: none"> ▪ 509 – Refresh Trade message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
ReferencePrice	20	4	Binary Signed	Indicative matching price
PairedQty	24	2	Binary	paired off quantity at the indicative matching price (Indicative matching quantity)
TotalImbalanceQty	26	2	Binary	total imbalance quantity at the indicative matching price.
MarketImbalanceQty	28	2	Binary	Total market order imbalance at the indicative matching price
AuctionTime	30	2	Binary	Projected Auction Time (hhmm)
AuctionType	32	1	ASCII	O (Opening) M (Market) H (Halt)
ImbalanceSide	33	1	ASCII	Side of the imbalance Buy/sell. Valid Values: <ul style="list-style-type: none"> ▪ 'B' – Buy ▪ 'S' – Sell ▪ Space – No imbalance
Reserved	34	2	Binary	Filler

7.6 REFRESH COMPLEX QUOTE MESSAGE STRUCTURE – MSG TYPE 511

The Refresh Complex Quotes message is sent in the event no quote messages are published for the symbol for 2 consecutive minutes. It has the same field content as the last published Complex Quote message, with the following exceptions:

- MsgType is 511
- SeriesSeqNum increments on every publication

This message is sent through the Complex Top of Book product.

Refresh Complex Quote Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 40 Bytes
MsgType	2	2	Binary	Type of message: <ul style="list-style-type: none"> 511 – Refresh Complex Quote Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
ComplexIndex	12	4	Binary	Numerical representation of the complex symbol.
SymbolSeqNum	16	4	Binary	Sequence number of messages for the complex symbol.
AskPrice	20	4	Binary Signed	Best Ask price. Should be used with the Price scale from the symbol mapping index.
BidPrice	24	4	Binary Signed	Best Bid price. Should be used with the Price scale from the symbol mapping index.
AskShares	28	2	Binary	Total quantity available at the above Ask price.
BidShares	30	2	Binary	Total quantity available at the above Bid price.
AskCustomerShares	32	2	Binary	Total quantity of 'Customer' orders available at the above Ask price.
BidCustomerShares	34	2	Binary	Total quantity of 'Customer' orders available at the above Bid price.
QuoteCondition	36	1	ASCII	1 (Regular Trading) 2 (Rotation) 3 (Trading Halted)
Reserved1	37	1	Binary	Filler
Reserved2	38	2	Binary	Filler

7.8 REFRESH COMPLEX TRADE MESSAGE STRUCTURE– MSG 513

The Refresh Complex Trade message is sent in the event no Complex Trade messages are published for the symbol for 2 consecutive minutes. It has the same field content as the last published Complex Trade message, with the following exceptions:

- MsgType is 513
- SeriesSeqNum increments on every publication

This message is sent through the Complex Top of Book product.

Refresh Trade Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: ▪ 32 bytes
MsgType	2	2	Binary	Type of message: Type of message: ▪ 513 – Refresh Complex Trade Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
ComplexIndex	12	4	Binary	Numerical representation of the symbol.
SymbolSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
TradeID	20	4	Binary	(blank) n/a for complex symbols
Price	24	4	Binary Signed	Price of the trade. Can be negative. Use the Price scale from the symbol mapping index.
Volume	28	2	Binary	Volume of the trade in actual number of contracts
TradeCond1	30	1	ASCII	Blank (regular trade) I (Late report) R (Floor trade) S (ISO sweep trade)
TradeCond2	31	1	ASCII	n/a

7.9 REFRESH COMPLEX IMBALANCE MESSAGE – MSG TYPE 515

The Refresh Complex Imbalance message is sent in the event no Complex Imbalance messages are published for the symbol for 2 consecutive minutes during Pre-opening or a symbol Halt. It has the same field content as the last published Complex Imbalance message, with the following exceptions:

- MsgType is 515
- SeriesSeqNum increments on every publication

This message is sent through the Complex Top of Book product.

Refresh Complex Imbalance Message fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: <ul style="list-style-type: none"> ▪ 36 bytes
Msg Type	2	2	Binary	Type of message. <ul style="list-style-type: none"> ▪ 515 – Refresh Complex Imbalance
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
ComplexIndex	12	4	Binary	Numerical representation of the symbol.
SymbolSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
ReferencePrice	20	4	Binary Signed	Indicative matching price
PairedQty	24	2	Binary	Paired off quantity at the indicative matching price (Indicative matching quantity)
TotalImbalanceQty	26	2	Binary	Total imbalance quantity at the indicative matching price.
MarketImbalanceQty	28	2	Binary	Total market order imbalance at the indicative matching price
AuctionTime	30	2	Binary	Projected Auction Time (hhmm)
AuctionType	32	1	ASCII	O (Opening) M (Market) H (Halt)
ImbalanceSide	33	1	ASCII	Side of the imbalance Buy/sell. Valid Values: <ul style="list-style-type: none"> ▪ 'B' – Buy ▪ 'S' – Sell ▪ Space – No imbalance
Reserved	34	2	Binary	Filler

7.10 REFRESH OUTRIGHT TRADE CORRECTION MESSAGE STRUCTURE – MSG TYPE 517

The Refresh Outright Trade Correction message is sent in the event that an Outright Trade Correction message is sent, followed by an interval of 2 minutes in which no Outright Trade, Correction or Cancel messages are published for the same symbol. It has the same field content as the last published Outright Trade Correction message, with the following exceptions:

- MsgType is 517
- SeriesSeqNum increments on every publication

This message is sent through the Outright Top of Book product.

Trade Correction Message Fields

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: <ul style="list-style-type: none"> 36 bytes
Msg Type	2	2	Binary	Type of message: <ul style="list-style-type: none"> 517 – Refresh Outright Trade Correction Message
SourceTime	4	4	Binary	Second portion of the ME timestamp.
SourceTimeNS	8	4	Binary	Nanosecond portion of the ME timestamp.
SeriesIndex	12	4	Binary	Numerical representation of the outright options symbol.
SeriesSeqNum	16	4	Binary	Sequence number of messages for the Outright options symbol.
OriginalTradeID	20	4	Binary	Original TradeID of the trade.
TradeID	24	4	Binary	New unique TradeID used to identify the corrected trade.
Price	28	4	Binary Signed	Price of the order. Use the Price scale from the symbol mapping index.
Volume	32	2	Binary	Volume of the trade in number of contracts.
TradeCond1	34	1	ASCII	Blank (regular trade) I (Late report) R (Floor trade) S (ISO sweep trade)
TradeCond2	35	1	ASCII	Complex indicator P (Complex trade with equity trade) L (Complex trade)