PILLAR COMMON CLIENT SPECIFICATION

NYSE PILLAR FEEDS

NYSE AMERICAN PILLAR FEEDS

NYSE ARCA PILLAR FEEDS

NYSE CHICAGO PILLAR FEEDS

NYSE NATIONAL PILLAR FEEDS

NYSE ARCA OPTIONS FEEDS (New)

NYSE AMERICAN OPTIONS FEEDS (New)

Version
2.6c

Date
August 3, 2021

Preface

DOCUMENT HISTORY

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

Version	Date	Change Description
2.4	Aug 28, 2020	CTA Regulatory Initiative - Updated Halt Reason Codes on msgtype 34 for primary securities - NYSE/Arca/American alongside Pillar implementation.
2.4a	Nov 20, 2020	Retransmission Request quotas increased from 5,000 to 10,000. Replace BATS w ith CBOE. Replaced XDP w ith proprietary data feed.
2.5	Dec 1, 2020	Added outright series support for NYSE Arca and American Options Removed section 5.1.7 Differences between Legacy RCF and Pillar Request Server for equity as equity migration is complete
2.6	March 5, 2021	Added complex series support for NYSE Arca and American Options Added PriceScaleCodeCabinet for Outright Series Index Mapping Messages Added options symbol mapping file section
2.6a	June 2, 2021	Removed PriceScaleCode Cabinet for Outright Series Index Mapping Messages and symbol mapping file. Corrected SecurityStatus ‘T’ to ‘B’
2.6b	July 27, 2021	Updated OptionSymbolRoot in Outright Series Index Mapping Message (Msg 50) from 5 to 6 char, includes offset modification. Updated PutOrCall field definition from ASCII to Binary in Outright Series Index Mapping Message (Msg 50) Added clarification on PriceScaleCode for Complex Series to match any of the underlying Outright Series (with a default value of ‘4’) Updated Complex Series Index Mapping message length from 80 to 109 characters
2.6c	Aug 3, 2021	Added Request Server Denial of Service section to Pillar Request Server Updated Max number of Refresh Requests in a day from 5000 to 10000

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:

- SFTI connectivity documents
- NYSE Symbology Specification
- IP Addresses

CONTACT INFORMATION

Service Desk

- Telephone: +1 212 896-2830
- Email: support@nyse.com
- Data Sales: datasales@nyse.com

FURTHER INFORMATION

- For additional product information please visit NYSE Real Time Market Data.
- For updated bandwidth , visit the capacity .

Contents

1. Introduction

1.1 RECEIVING REAL TIME MARKET DATA

Real-time PILLAR 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.

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 each PILLAR feed can be found at https://www.nyse.com/publicdocs/nyse/data/IP_Addresses.xls. A client application receives a product by subscribing to some or all of the channels that make up the feed.

In response to requests for retransmission and refresh, market 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 Pillar Request Server for complete information.

2. Packets and Heartbeats

2.1 PACKET HEADER

All packets sent on any PILLAR feed have an PILLAR Packet Header followed by one or more messages (with the exception of Heartbeat packets which 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 Packet Header Structure

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
PktSize	0	2	Binary	The size of the packet in bytes, including this 16 -byte packet header
DeliveryFlag	2	1	Binary	A flag that indicates whether this is an original, retransmitted, or 'replayed' message. Valid values include: ▪ 1 – Heartbeat ▪ 10 – PILLAR Failover (see PILLAR Publisher Failover) ▪ 11 – Original Message ▪ 12 – Sequence Number Reset Message ▪ 13 – Only one packet in retransmission sequence ▪ 15 – Part of a retransmission sequence ▪ 17 – Only one packet in Refresh sequence ▪ 18 – Start of Refresh sequence ▪ 19 – Part of a Refresh sequence ▪ 20 – End of Refresh sequence ▪ 21 – Message Unavailable
NumberMsgs	3	1	Binary	The number of messages in this packet
SeqNum	4	4	Binary	The message sequence number of the first message in this packet
SendTime	8	4	Binary	The time when this packet was published to the multicast channel, in seconds since Jan 1, 1970 00:00:00 UTC.
SendTimeNS	12	4	Binary	The nanosecond offset from the Send Time

2.2 HEARTBEATS

To assist the client in confirming connection health, application heartbeats are sent once a minute by the Request Server, and once a second by the real-time publishing servers (data, refresh and retransmissions channels).

A heartbeat consists of a packet containing a Packet Header and no messages. The Packet Header's Delivery Flag is set to 1 and Number Msgs is 0. Since a heartbeat packet contains no messages, a heartbeat does not increment the next expected sequence number. See Sequence Numbers.

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

3. Message Field Content

Messages are contiguous data structures consisting of fixed-length fields. No names or 'tags' appear in the message.

- Message fields align on 1 byte boundaries, so there are no filler fields for alignment purposes.
- Binary fields are published in Little-Endian ordering.
- Binary integer values are unsigned unless otherwise specified.
- All ASCII string fields are left aligned and null padded.
- Segmentation of messages across packets is not supported, so a message will never straddle a packet boundary.
- The length of a message as actually published may differ from the length of the message structure defined in the client specifications. See Msg Size Field below for details.

3.1 MESSAGE HEADER

The format of each message varies according to type, but each type starts with a standard 4-byte message header:

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	The size of this message in bytes
MsgType	2	2	Binary	The type of this message

3.1.1 Msg Size Field

In order to handle future releases of PILLAR feeds smoothly, clients should never hard code msg sizes in feed handlers. Instead, the feed handler should use the Msg Size field to determine where the next message in a packet begins.

This allows

- Support of PILLAR format variations among markets
- Client flexibility when revised message structures go live in production

In example 1 below, a message type is defined in the specification to have different lengths in different markets. The trailing field is not published in the Arca market. An Arca-coded client can process NYSE data correctly (but of course cannot use the trailing Volume field without field-specific coding).

Example 1: Message type with format variations across markets

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message. **NYSE – 24 bytes** **NYSE American – 24 bytes** **NYSE Arca - 20 bytes**
Msg Type	2	2	Binary	The type of this message: 998 – Example 1 msg type
SourceTimeNS	4	4	Binary	
SymbolIndex	8	4	Binary	
OrderID	12	4	Binary	
Price	16	4	Binary	
Volume	28	4	Binary	Not published in Arca market

Look at the Msg Size field to know where the next message starts.

Market-specific content

The variable message size can also insulate client code from future field additions that you may not need.

In example 2, an existing message type is 16 bytes long.

Example 2: Release N

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: 16 bytes
Msg Type	2	2	Binary	The type of this message: 999 – Price message example
SourceTimeNS	4	4	Binary	
SymbolIndex	8	4	Binary	
Price	12	4	Binary	

Look at the Msg Size field to know where the next message starts.

In a future release, a four-byte volume field will be added, increasing the Msg Size to 20 bytes.

If the client wishes to delay upgrading his feed handler for the new content, no coding is needed at the time of the release. Proper coding of the MsgSize field up front allows the client to handle the unforeseen 20-byte format. On his own schedule, the client can upgrade his feed handler to process the new field.

Example 2: Release N+1: a new field is added

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: 20 bytes
Msg Type	2	2	Binary	The type of this message: 999 – Price message example
SourceTimeNS	4	4	Binary	
SymbolIndex	8	4	Binary	
Price	12	4	Binary	
Volume	16	4	Binary	New field

Look at the Msg Size field to know where the next message starts.

Unmodified clients can handle longer message structure (but can't benefit from new content)

3.2 DATE AND TIME CONVENTIONS

Dates and times are in UTC (Universal Time, Coordinated), and are expressed in nanoseconds since the Unix Epoch (Jan 1, 1970 00:00:00). A complete timestamp consists of two 4-byte fields: seconds since the Unix Epoch, and nanoseconds within the current second, as in a Unix timespec structure.

The PILLAR Packet Header contains SendTime and SendTimeNS fields to show the time that the packet was published to the wire by the PILLAR Publisher.

Most PILLAR messages additionally contain a timestamp called Source Time to show the time of the Matching Engine event that caused the publication of this message.

Many of the higher-volume PILLAR feeds such as Integrated and BBO explicitly publish only the nanoseconds portion of the Source Time in each message. The seconds portion is explicitly published in a Source Time Reference Message (Msg Type 2) once a second.

Source Time Reference messages are published per Matching Engine partition (per TXN, which is equivalent to the Integrated Feed channel number).

3.3 SEQUENCE NUMBERS

Each message in a given channel is assigned a unique sequence number. Sequence numbers increase monotonically per channel, and can be used to detect publication gaps.

To optimize publication efficiency, the sequence number is not explicitly published in each message. Instead, the Packet Header contains the sequence number of the first message in the packet, along with the number of messages in the packet. Using these fields, the client can easily associate the correct sequence number with each message.

The sequence number combined with the channel ID form a message ID which is unique across the feed.

3.4 SYMBOL SEQUENCE NUMBERS AND SERIES SEQUENCE NUMBERS

In addition to the sequence number, many message types explicitly include a field called Symbol Sequence Number (Series Sequence Number for Options), which identifies the message's position in the sequence of all messages published by the feed for a given symbol.

Clients who are tracking only a small number of symbols may opt to ignore sequence numbers and track only Symbol Sequence Numbers (Series Sequence Number for Options) for each symbol or series of interest. If such a client ever experiences a Symbol Sequence Number (Series Sequence Number for Options) gap, he can request a refresh for that symbol or series.

3.5 PRICES

All price fields are published as signed binary integers. To interpret a price correctly, the client must use the published price value as a numerator along with the Price Scale Code in the symbol's Symbol Index Mapping Message (Msg Type 3), Outright Series Index Mapping Message (Msg Type 50) and Complex Series Index Mapping Message (Msg Type 60) as follows:

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

For example, a price of $27.56 is represented as a published price field of 2756 and a PriceScaleCode of 2.

3.6 ORDER ID'S AND TRADE ID'S

The Order ID and the Trade ID in order-based feeds such as NYSE Integrated Feed are binary integers that uniquely identify an order or an execution. Order and Trade IDs are valid for the trading day only.

3.6.1 Standard Pillar Correlation Rules

These rules, which assume use of a Pillar matching engine and a Pillar Gateway, are applicable to the all Pillar Equity and Options markets.

Order IDs are 8 bytes long and correlate uniquely across markets to the 8 byte OrderID in the gateway Order Ack.

Trade IDs are 4 bytes long and correspond to the lowest 4 bytes of the 8-byte Deal ID in the gateway Execution Report. The Trade ID is unique per ME symbol partition (System ID in the Symbol Index Mapping message), and therefore unique per symbol. Prepend 3 fields to the Trade ID as discussed below to make a unique match across markets to the Deal ID field in the gateway Execution Report.

3.6.1.1 Correlating ID fields in the market data with fields in the order entry API

By combining a 4-byte Order ID or Trade ID from an Integrated Feed message with the Market ID and System ID from the Symbol Mapping message as shown below, you can obtain the corresponding 8-byte ID from the gateway API.

The table assumes the client byte ordering is Little Endian. If the client byte ordering is Big Endian, the byte order is reversed.

PILLAR FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
	0	1	Binary	0

PILLAR FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
System ID	1	1	Binary	Unique ID for a single matching engine instance (Pillar Symbol Partition or UTP TU) found in the Symbol Index Mapping message's System ID field
Market ID	2	2	Binary	ID of the originating market in the Symbol Index Mapping message
OrderID or TradeID	4	4	Binary	Contents of 4-byte field being disambiguated

3.6.2 NYSE Arca Options and NYSE American Options Transition to Pillar

As of this publication, NYSE Arca Options and NYSE American Options trade on legacy UTP matching engine technology. All order/quote entry is via legacy CGW, MMD, and UGW gateways.

NYSE Arca and NYSE American Options will migrate to Pillar on a published schedule with an underlying-based roll out.

When Pillar Gateways are made available, they will support order/quote entry for series that have been migrated to Pillar and will not be made available in parallel for series traded on UTP. Similarly, legacy GWs will not support order/quote entry for series trading on Pillar. The Standard Pillar correlation rules described in this document will only apply Series trading on Pillar and depend on the roll out schedule.

3.7 SYMBOL AND SERIES INDEXES

In all PILLAR feeds, symbol-specific referential data is published in a Symbol Index Mapping Message (Msg Type 3),for options outright series-specific referential data is published in a Outright Series Index Mapping Message (Msg 50) and for options complex series-specific referential data is published in a Complex Series Index Mapping Message (Msg 60) at system startup. Symbol and Series Index Mapping messages appear in each channel only for the symbols or outright/complex series that appear in that channel.

Any client who misses this initial spin can request a refresh of either Symbol or Outright Series Indexes by sending a Symbol Index Mapping Request Message (Msg Type 13)to the Request Server. The requested Symbol Index Mapping messages and Outright Series Mapping messages will be re-published over the Refresh channels. See Common Client Spec for info on the request server process.

The Symbol and Series Index Mapping messages include the ASCII symbol in NYSE format along with a unique ID called a Symbol Index (Outright Series Index for outright, Complex Series Index for complex). Other symbol or seriesspecific messages such as Trade and BBO messages contain only the Symbol Index (Outright Series Index for outright series, Complex Series Index for complex) and no other referential data.

Symbol and Series Indexes are the same for each symbol or outright/complex series every day and the same across all Pillar-powered NYSE equity and option markets.

If more than one Symbol and Series Index Mapping message is received for the same symbol within a trading day, the correspondence between the Symbol and the Symbol Index will not change, but other field values might. In this case, the latest field values override any earlier values, but do not apply retroactively.

4. Messages Sent by the Publisher

4.1 SYMBOL INDEX MAPPING MESSAGE (MSG TYPE 3)

This message is published over the real-time data channels at system startup or in the context of a refresh sequence after a Matching Engine or PILLAR Publisher failover. It provides referential data for a single specified symbol or underlying symbol.

See Symbol Indexes for more information.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: 44 bytes
Msg Type	2	2	Binary	The type of this message: 3 – Symbol Index Mapping Message
SymbolIndex	4	4	Binary	The unique ID of this symbol for all products within this market.
Symbol	8	11	ASCII	Null-terminated ASCII symbol in NYSE Symbology.
Reserved	19	1	Binary	This field is reserved for future use
Market ID	20	2	Binary	ID of the Originating Market: • 1 - NYSE Equities • 3 – NYSE Arca Equities • 4 – NYSE Arca Options • 8 – NYSE American Options • 9 - NYSE American Equities • 10 - NYSE National Equities • 11 - NYSE Chicago
System ID	22	1	Binary	ID of the Originating matching engine server.
Exchange Code	23	1	ASCII	For listed equity markets, the market where this symbol is listed: ▪ A – NYSE American ▪ L - LTSE ▪ N – NYSE ▪ P – NYSE Arca ▪ Q – NASDAQ ▪ V - IEX ▪ Z – CBOE
PriceScaleCode	24	1	Binary	Specifies placement of the decimal point in price fields for this security. See Prices .
Security Type	25	1	ASCII	Type of Security used by Pillar-powered markets ▪ A – ADR ▪ C - COMMON STOCK ▪ D – DEBENTURES ▪ E – ETF ▪ F – FOREIGN ▪ H – US DEPOSITARY SHARES ▪ I – UNITS ▪ L – INDEX LINKED NOTES ▪ M - MISC/LIQUID TRUST ▪ O – ORDINARY SHARES ▪ P - PREFERRED STOCK

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
				▪ R – RIGHTS ▪ S - SHARES OF BENEFICIARY INTEREST ▪ T – TEST ▪ U – CLOSED END FUND ▪ W – WARRANT
Lot Size	26	2	Binary	Round lot size in shares.
PrevClosePrice	28	4	Binary	The previous day's closing price for this security.
PrevCloseVolume	32	4	Binary	The previous day's closing volume for the security.
Price Resolution	36	1	Binary	▪ 0 - All Penny ▪ 1 - Penny/Nickel ▪ 5 - Nickel/Dime
Round Lot	37	1	ASCII	Round Lots Accepted: ▪ Y – Yes ▪ N – No
MPV	38	2	Binary	For equity symbols, the minimum increment for a trade price, in 100ths of a cent. Typically 1, or $0.0001, but for some Tick Pilot stocks, can be 500, or $0.05 For underlying symbols, this field is default as blank.
Unit of Trade	40	2	Binary	For equity symbols, this field specifies the security Unit of Trade in shares. Valid values are 1, 10, 50 and 100 For underlying symbols, this field is default as blank.
Reserved	42	2	Binary	Reserved for future use.

4.2 OUTRIGHT SERIES INDEX MAPPING MESSAGE (MSG 50)

This message is published over the real-time data channels at system startup or in the context of a refresh sequence after a Matching Engine or PILLAR Publisher failover. It provides referential data for a single specified options outright series.

See Series Indexes for more information.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: 55 bytes
Msg Type	2	2	Binary	The type of this message: 50 – Outright Series Index Mapping Message
Series Index	4	4	Binary	The unique ID of this series for all products within this market.
Series Type	8	1	Binary	Identifies series type: • 0 - Standard • 1 - Flex • 2 - Flex percentage
Market ID	9	2	Binary	Identifies originating market:

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
				• 4 (NYSE Arca Options) • 8 (NYSE American Options)
System ID	11	1	Binary	ID of the originating matching engine server
OptionSymbolRoot	12	6	ASCII	Root symbol for options in OCC symbology (EX. BRKB)
UnderlyingSymbol	18	11	ASCII	Underlying symbol for options in NYSE symbology (EX. BRK B)
UnderlyingIndex	29	4	Binary	Underlying stock mapping index
PriceScaleCode	33	1	Binary	Price Scale Code for price conversion of the series. Default Series Price Scale Code is '4'
ContractMultiplier	34	2	Binary	Number of Underlying shares per option contract
MaturityDate	36	6	ASCII	Option maturity date - YYMMDD
PutOrCall	42	1	Binary	Valid values: • 0 (Put) • 1 (Call)
StrikePrice	43	10	ASCII	Strike price. ASCII 0-9 with optional decimal point. EG:51.75, 123 (Option only)
ClosingOnlyIndicator	53	1	ASCII	Valid values: • 0 - Standard Series • 1 - Closing Only Series
Reserved	54	1	Binary	Reserved for future use

4.3 COMPLEX SERIES INDEX MAPPING MESSAGE (MSG 60)

This message is published over the real-time data channels at system startup or in the context of a refresh sequence after a Matching Engine or PILLAR Publisher failover. It provides referential data for a single specified options complex series.

On system startup, each Complex feed multicast channel sends all the symbols on its channel. Symbol Index Mapping messages are published first, followed by all Outright Series Index Mapping messages, and lastly in the Complex feed, Complex Series Index Mapping messages. This message is also sent intraday whenever a new complex symbol is created

See Series Indexes for more information.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
Msg Size	0	2	Binary	Size of the message: Variable, 109 bytes maximum
Msg Type	2	2	Binary	Type of message: - 60 - Complex Series Index Mapping
Series Index	4	4	Binary	The unique ID of this Complex series within this market.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
Market ID	8	2	Binary	Identifies originating market: • 4 (NYSE Arca Options) • 8 (NYSE American Options)
System ID	10	1	Binary	ID of the Originating matching engine server.
NoOfLegs	11	2	Binary	Number of legs in complex symbol 2 - 12
SymbolIndex	13	4	Binary	This field will repeat for each leg • Series index if Security type is Option • Underlying index if Security type is Equity
Leg Ratio Qty	17	2	Binary	Leg Ratio. This filed will repeat for each leg The maximum ratio between the smallest component leg quantity to the largest leg component quantity cannot exceed 3:1 or 1:3
Side	19	1	ASCII	Leg side. This filed will repeat for each leg • B (Buy) • S (Sell)
Security Type	20	1	ASCII	Leg Security Type. This filed will repeat for each leg • O (Options Series leg) • E (Equity stock leg)

Note: Complex Series Index Mapping Message does not have a PriceScaleCode as it matches the PriceScaleCode of any of the underlying Outright Series making up the Complex Series (see Msg 50) and default value is '4'.

4.4 SECURITY STATUS MESSAGE (MSG TYPE 34)

This message informs clients of changes in the status of a specific security or underlying symbol, such as Trading Halts, Short Sale Restriction state changes, etc.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 46 Bytes
MsgType	2	2	Binary	The type of this message: 34 – Security Status Message
SourceTime	4	4	Binary	The time when this msg was generated in the order book, in seconds since Jan 1, 1970 00:00:00 UTC.
SourceTimeNS	8	4	Binary	The nanosecond offset from the SourceTime
SymbolIndex	12	4	Binary	The unique ID of this symbol for all products within this market.
SymbolSeqNum	16	4	Binary	The unique ID of this message in the sequence of messages published for this specific symbol.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
Security Status	20	1	ASCII	The new status that this security is transitioning to. The following are Halt Status Codes: ▪ 4 - Trading Halt ▪ 5 - Resume ▪ 6 - Suspend The following are Short Sale Restriction Codes (published for all symbols traded on this exchange): ▪ A – Short Sale Restriction Activated (Day 1) ▪ C – Short Sale Restriction Continued (Day 2) ▪ D - Short Sale Restriction Deactivated Market Session values : ▪ P – Pre-opening ▪ B - Begin Accepting orders ▪ E – Early session ▪ O – Core session ▪ L – Late session ▪ X – Closed If this security is not halted at the time of a session change, the Halt Condition field = ~. If this security is halted on a session change, Halt Condition is non-~, and the security remains halted into the new session. The following values are the Price Indication values : ▪ I – Halt Resume Price Indication ▪ G – Pre-Opening Price Indication
Halt Condition	21	1	ASCII	▪ ~ - Security not delayed/halted ▪ D - News released ▪ I - Order imbalance ▪ P - News pending ▪ M – LULD pause ▪ X - Equipment changeover ▪ A - Additional Information Requested ▪ C - Regulatory Concern ▪ E - Merger Effective ▪ F - ETF Component Prices Not Available ▪ N - Corporate Action ▪ O - New Security Offering ▪ V - Intraday Indicative Value Not Available Market Wide Circuit Breakers: ▪ 1 - Market Wide Circuit Breaker Halt Level 1 ▪ 2 - Market Wide Circuit Breaker Halt Level 2 ▪ 3 - Market Wide Circuit Breaker Halt Level 3
Reserved	22	4	Binary	Reserved for future use.
Price 1	26	4	Binary	Default value is 0. ▪ If securityStatus = A and this security is listed on this exchange, then this field is the SSR Triggering Trade Price ▪ If securityStatus = G or I, then this field is the Indication Low Price.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
Price 2	30	4	Binary	Default value is 0 ▪ If securityStatus = G or I, then this field is the Indication High Price.
SSR Triggering Exchange ID	34	1	ASCII	This field is only populated when securityStatus = A and this security is listed on this exchange. Otherwise it is defaulted to 0x20. Valid values are: ▪ A – NYSE American ▪ B – NASDAQ OMX BX ▪ C – NYSE National ▪ D – FINRA ▪ I – ISE ▪ J – CBOE EDGA ▪ K – CBOE EDGX ▪ L - LTSE ▪ M – NYSE Chicago ▪ N – NYSE ▪ P – NYSE Arca ▪ Q – NASDAQ ▪ S – CTS ▪ T – NASDAQ OMX ▪ V – IEX ▪ W – CBSX ▪ X – NASDAQ OMX PSX ▪ Y – CBOE BYX ▪ Z – CBOE BZX ▪ H - MIAX ▪ U - MEMX
SSR Triggering Volume	35	4	Binary	Default value is 0. This field is only populated when securityStatus = A and this security is listed on this exchange
Time	39	4	Binary	Default value is 0. Format : HHMMSSmmm (mmm = milliseconds) ▪ If securityStatus = A and this security is listed on this exchange, then this field is the SSR Trigger Time
SSRState	43	1	ASCII	The current SSR state, which this msg updates if the Security Status field contains an SSR Code. Valid values: ▪ ~ – No Short Sale Restriction in Effect ▪ E – Short Sale Restriction in Effect
MarketState	44	1	ASCII	The current Market State, which this msg updates if the Security Status field contains a Market State Code. Valid values: ▪ P – Pre-opening ▪ E – Early session ▪ O – Core session ▪ L – Late session (Non-NYSE only) ▪ X -- Closed
SessionState	45	1	ASCII	Unused. Defaulted to 0x00.

4.5 OPTIONS STATUS MESSAGE (MSG TYPE 51)

This message informs clients of changes in the status of a specific option outright series and complex series, such as Trading Halts etc.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 23 Bytes
MsgType	2	2	Binary	The type of this message: 51 – Options Status Message
SourceTime	4	4	Binary	The time when this message was generated in the order book, in seconds since Jan 1, 1970 00:00:00 UTC.
SourceTimeNS	8	4	Binary	The nanosecond offset from the SourceTime
SeriesIndex	12	4	Binary	The unique ID of this series within this market.
SeriesSeqNum	16	4	Binary	The unique ID of this message in the sequence of messages published for this specific series.
Series Status	20	1	ASCII	The new status that this series is transitioning to. The following are Halt Status Codes: ▪ 4 - Trading Halt ▪ 5 - Resume ▪ 6 - Suspend Market Session values : ▪ P – Pre-opening ▪ B - Begin Accepting orders ▪ O – Core session ▪ X – Closed If this series is not halted at the time of a session change, the Halt Condition field = ~. If this series is halted on a session change, Halt Condition is non-~, and the series remains halted into the new session.
Market State	21	1	ASCII	Market Session values : • P – Pre-opening • O – Core session • X – Closed
Halt Condition	22	1	ASCII	▪ ~ - series not delayed/halted ▪ h - Option series is halted

4.6 SEQUENCE NUMBER RESET MESSAGE (MSG TYPE 1)

This message is sent to reset the Message Sequence Number at start of day, or in response to failure recoveries.

This message always appears in its own dedicated packet with a Sequence Number of 1 (the new, reset number). The packet Delivery Flag is normally 12, as on system startup.During failover events the flag is set to 10.

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTIO
MsgSize	0	2	Binary	Size of the message: 14 Bytes
MsgType	2	2	Binary	The type of this message: 1 – Sequence Number Reset message
SourceTime	4	4	Binary	The time when this msg was generated in the order book, in seconds since Jan 1, 1970 00:00:00 UTC.
SourceTimeNS	8	4	Binary	The nanosecond offset from the SourceTime
ProductID	12	1	Binary	The unique ID for this NYSE feed listed in the feed's client specification.
ChannelID	13	1	Binary	The ID of the multicast channel over which the packet was sent.

4.7 SOURCE TIME REFERENCE MESSAGE (MSG TYPE 2)

This message is sent at the start of every second during periods of active data publication. Unlike some control messages, Source Time Reference messages can come in packets containing market data messages.

The client can concatenate the SourceTime field with the SourceTimeNS field in subsequent market data messages to get full 8-byte Matching Engine event timestamps. The contents of the ID field can be linked via the Symbol Index Mapping Message (Msg Type 3) to the applicable data messages.

See Date and Time Conventions for more information.

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 16 bytes
MsgType	2	2	Binary	The type of message: 2 – Source Time Reference Message
ID	4	4	Binary	ID of the originating Matching Engine partition to which this message applies. This usage will become standard across all products in future releases.
SymbolSeqNum	8	4	Binary	Reserved for future use. Ignore any content. This usage will become standard across all products in future releases.
SourceTime	12	4	Binary	The time when this msg was generated in the order book, in seconds since Jan 1, 1970 00:00:00 UTC.

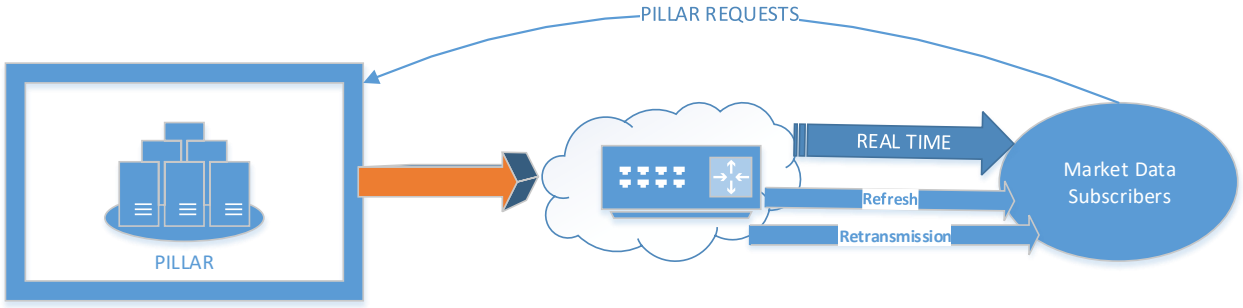
4.8 SYMBOL CLEAR MESSAGE (MSG TYPE 32)

In case of a failure and recovery of a Matching Engine or an PILLAR Publisher, the publisher may send a full state refresh for every symbol or series affected. This kind of unrequested refresh is preceded by a Symbol Clear message. The client should react to receipt of a Symbol Clear message by clearing all state information for the specified symbol or series in anticipation of receiving a full state refresh.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 20 Bytes
MsgType	2	2	Binary	The type of this message: 32 – Symbol Clear
SourceTime	4	4	Binary	The time when this msg was generated in the order book, in seconds since Jan 1, 1970 00:00:00 UTC.
SourceTimeNS	8	4	Binary	The nanosecond offset from the SourceTime
SymbolIndex	12	4	Binary	The unique ID of this symbol or series for all products within this market.
NextSourceSeqNum	16	4	Binary	The sequence number in the next message for this symbol

5. Error Handling via the Pillar Request Server

5.1 PILLAR REQUEST SERVER



Similar to the NYSE Pillar Gateway, the new Pillar Request Server will facilitate market data client requests for Refresh/ Retransmission with the following features:

- In case of dropped multicast packets, the client can connect to the Pillar 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 Pillar 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, clients can connect to the Pillar Request Server and request a refresh of some or all of the missed data.
- This service is subject to Pillar's 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 Pillar Request Server after the client has connected a truly excessive number of times.

Customers are required to certify their market data sessions for refresh/retransmission in the NYSE Pillar CERT environment before activation in production.

5.1.1 Request Processing

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.

While it is possible to connect to the Request Server only as needed, and disconnecting after each request, it is recommended that Customers remain connected to the Request Server for the entire trading day.

5.1.1.1.1 Handling Sequence Number Gaps

Since multicast is an unreliable protocol, messages can be dropped. For this reason, clients are advised to process both lines in a channel. If a 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 10) via TCP to the Request Server. The Retransmission Request contains a unique client ID called a Source ID, along with the Product and Channel IDs and the sequence number range of the missing messages.

On receipt of a Retransmission Request message, the Request Server will send back a Request Response Message (Msg Type 11). If any of the fields of the the Retransmission Request contained malformed or meaningless information, the request is rejected. If the request is accepted, the Retransmission Server will re-send the requested messages via multicast over the Retransmission channels.

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.

5.1.1.2 Request Quotas

The table below summarizes the retransmission/refresh request limitations that are enforced by the Pillar Request Server. The numbers represent thresholds per channel.

FEATURE	DESCRIPTION
Max number of packets per Retransmission Request	Retransmission requests for more than 1,000 messages will not be honored.
Max number of Retransmission Requests in a day	Retransmission requests from a client who has already made 10,000 retransmission requests today will not be honored.the clientid will be blocked from making retransmission requests for the remainder of the day.
Max number of Refresh Requests in a day	Refresh requests from a client who has already made 10,000 refresh requests today will not be honored.

5.1.2 Request Server Denial of Service

In the event a client attempts to connect but fails, or sends malformed messages, the Pillar Denial of Service (DOS) functionality will be triggered on the Pillar Request Server. Operational Parameters:

- 100 failed login attempts
- 100 retransmission/refresh request rejects

On a subsequent connection event, Pillar Request Server will lockout the client id for 60 seconds.

If the firm's client id continues to exhibit DOS behaviors, NYSE operations may shutdown the offending retransmission port for the day. In this event, the firm should reach out to NYSE connectivity support, connectivity@nyse.com.

5.1.3 Retransmission Format

Retransmitted messages have the same message format and content as the originally published messages (including the Sequence Numbers, but they may be packetized differently for best efficiency.

Packets of retransmitted messages have special Delivery Flag values in the Packet Header:

- 13 – Only one packet in retransmission sequence
- 15 – Part of a retransmission sequence

5.1.4 Recovering from Client Late Starts or Intraday Failures

If a client process 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 accomplish this, the client can request a refresh from the Pillar Request Server.

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 the Pillar Request Server. This connection should be maintained all day.
3. Subscribe to the Refresh multicast channels.
4. Send a Refresh Request Message (Msg Type 15) to the Request Server.

The Refresh Request contains:

1. A unique client ID called a Source ID
2. Product and Channel IDs
3. A Symbol Index, specifying a particular symbol to be refreshed or else if 0, specifying all symbols.

On receipt of a Refresh Request message, the Request Server will send back a Request Response Message (Msg Type 11). If any of the fields of the 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. Session related Quota can be replenished based on User/Client Id Level flag "replenish". If further requests are required, please contact NYSE.

If the request is accepted, the Pilar Request Server will send the snapshot message(s) over the specific Refresh channel. 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 sequence numbers are lower than the LastSequenceNumber indicated in the refresh sequence should be discarded.

5.1.5 Refresh Message Format

Each refresh packet begins with a Packet Header, followed by a Refresh Header (Msg Type 35).

The Packet Header for a refresh packet has special Delivery Flag values:

- 17 – Only one packet in Refresh sequence
- 18 – Start of Refresh sequence
- 19 – Part of a Refresh sequence
- 20 – End of Refresh sequence

The Refresh Header identifies the position of the current packet is in this sequence of Refresh packets, along with the total number packets in this sequence. By use of the Delivery Flag and the packet sequence information in the Refresh Header, the client can know when the last packet of the refresh sequence has been received.

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

5.1.6 Refreshing Symbol Information

At system startup, each channel publishes a Symbol Index Mapping Message (Msg Type 3) for each symbol published on this channel.

If a client process misses the initial spin of symbol information for whatever reason, he may wish to receive a refresh of some or all Symbol Index Mapping messages before resuming processing of real-time data. To do this, the client should follow the procedure described in Recovering from Client Late Starts or Intraday Failures, but send a Symbol Index Mapping Request Message (Msg Type 13) to the Request Server instead of a Refresh Request Message.

5.1.7 Symbol Index Mapping Refresh Format

Requested Symbol Index Mapping messages are published by the Refresh Server with the same Packet Header Delivery Flags used for Refresh publications. Refresh Headers are not used in Symbol Index Mapping refreshes.

6. Pillar Request Server - Client Request Message

6.1 RETRANSMISSION REQUEST MESSAGE (MSG TYPE 10)

Clients who have experienced a sequence number gap and need a retransmission of the missed messages should send a Retransmission Request message via TCP to the Request Controller. A Request Response message will be sent over the TCP connection back to the client, and if the request was valid, the requested message(s) will be re-published over the relevant Retransmission multicast channel.

The retransmitted message(s) will have the same message format and content as the original messages that were missed.

Retransmission Requests should be sent in a packet whose Packet Header Delivery Flag is set to 11, and which contains a valid sequence number.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 24 Bytes
MsgType	2	2	Binary	The type of this message: 10 – Retransmission Request message
BeginSeqNum	4	4	Binary	The beginning sequence number of the range of messages to be retransmitted.
EndSeqNum	8	4	Binary	The end sequence number of the range of messages to be retransmitted.
SourceID	12	10	ASCII	The ID of the client requesting this retransmission . All trailing characters should be NULL. Examples: • 10 character: ABCDEFGHIJ • 9 character: ABCDEFGHI<NULL>
ProductID	22	1	Binary	The unique ID of the feed for which a retransmission is requested (listed in the feed's client specification).
ChannelID	23	1	Binary	The ID of the multicast channel on which the gap occurred.

7. Refresh and Retransmission - Client Response Messages

7.1 REFRESH HEADER (MSG TYPE 35)

The first message in each packet of refresh messages published over the Refresh multicast channels is of this type.

Valid values for the DeliveryFlag in the PacketHeader are:

- 17 – Only one packet in Refresh sequence
- 18 – Start of Refresh sequence
- 19 – Part of a Refresh sequence
- 20 – End of Refresh sequence
-

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 16 Bytes
MsgType	2	2	Binary	The type of this message: 35 – Refresh Header Message
CurrentRefreshPkt	4	2	Binary	The current refresh packet in the update
TotalRefreshPkts	6	2	Binary	The total number of refresh packets you should expect in the update
LastSeqNum	8	4	Binary	The last sequence number sent on the channel for any symbol. The refresh is the state of the order book as of this sequence number.
LastSymbolSeqNum	12	4	Binary	The last symbol sequence number sent for this symbol. The refresh is the symbol state of this symbol as of this symbol sequence number.

7.1.1 Shortened Refresh Header

The first message in the first packet for a given symbol is a full 16-byte Refresh Header message.

Every other packet for the same symbol contains an 8-byte Refresh Header. The LastSeqNum and the LastSymbolSeqNum fields are removed so as not to send duplicate information in every packet.

7.1.2 Refresh Example

Assuming this refresh of a single symbol requires three packets:

The first, second and third Packet structures look as follows:

PACKET HDR delivery = first	FULL REFRESH HDR	MESSAGE 1	MESSAGE 2	...	MESSAGE N
PACKET HDR delivery = part	SHORT REFRESH HDR	MESSAGE 1	MESSAGE 2	...	MESSAGE N
PACKET HDR delivery = last	SHORT REFRESH HDR	MESSAGE 1	MESSAGE 2	...	MESSAGE N

For a depth of book feed such as PILLAR Integrated or PILLAR ArcaBook, the sequence of refresh messages per symbol consists of the following message types:

1. Symbol Index Mapping Message (Msg Type 3)
2. Imbalance Message (Msg Type 105), if there is a current imbalance
3. Security Status Message (Msg Type 34)

4. Add Order Refresh (Msg Type 106), repeated as needed to specify the book state for this symbol

7.1.3 Header Fields in the Refresh Channels

7.1.3.1 Refresh response to a request for all Symbol Index Mapping messages

There are no Refresh Header messages

- First packet Delivery Flag =18 (START of refresh)
- Intermediate packets Delivery Flag = 19 (PART of refresh)
- Last packet Delivery Flag = 20 (END of refresh)

7.1.3.2 Refresh response to a request for a single Symbol Index Mapping message

There is no Refresh Header message.

- One packet is sent Delivery Flag = 17 (ONE packet in the refresh)

7.1.3.3 Refresh response to a request for a full refresh of all symbols

Each packet contains messages for a single symbol only.

- All packets for the first symbol Delivery Flag =18 (START of refresh)
- All packets for intermediate symbols Delivery Flag = 19 (PART of refresh)
- All packets for the last symbol Delivery Flag = 20 (END of refresh)

The first message in each packet is a Refresh Header.

For each symbol:

- The currentRefreshPkt and totalRefreshPkts fields in the Refresh Header apply to this symbol only.
- The first packet contains a full Refresh Header (16 bytes). The LastSequenceNumber field contains the sequence number of the last message processed in this channel for any symbol. The LastSymbolSeqNum field contains the last Symbol Sequence Number processed for this symbol.
- All subsequent packets contain a short Refresh Header (8 bytes).

7.1.3.4 Refresh response to a request for a full refresh of a single symbol

- If there are multiple packets in the response Delivery Flags = 19 (PART of refresh)
- If there is only one packet in the response Delivery Flag = 17 (ONE packet in the refresh sequence)

All packets begin with a Refresh Header message.

- The first packet contains a full Refresh Header (16 bytes).
- The first packet for a symbol contains a full Refresh Header (16 bytes). The LastSequenceNumber field contains the sequence number of the last message processed in this channel for any symbol. The LastSymbolSeqNum field contains the last Symbol Sequence Number processed for this symbol.
- All subsequent packets contain a short Refresh Header (8 bytes).

7.2 « MESSAGE UNAVAILABLE » MESSAGE (MSG TYPE 31)

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

For any packet containing a Message Unavailable message, the Packet Header Delivery Flag will be set to 21.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 14 Bytes
MsgType	2	2	Binary	The type of this message: 31 – Message Unavailable
BeginSeqNum	4	4	Binary	The beginning sequence number of the unavailable range of messages.
EndSeqNum	8	4	Binary	The ending sequence number of the unavailable range of messages.
ProductID	12	1	Binary	The unique ID of the feed for which the retransmission was requested (listed in the feed's client specification).
ChannelID	13	1	Binary	The ID of the multicast channel for which the retransmission was requested.

7.3 REFRESH REQUEST MESSAGE (MSG TYPE 15)

Clients who have experienced a failure and need a refresh of the state of one or all symbols in a specific channel should send a Retransmission Request message via TCP to the Request Controller. A Request Response message will be sent over the TCP connection back to the client, and if the request was valid, the requested message(s) will be published over the relevant Refresh multicast channel.

Retransmission Requests should be sent in a packet whose Packet Header Delivery Flag is set to 11, and which contains a valid sequence number.

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 20 Bytes
MsgType	2	2	Binary	The type of this message: 15 – Refresh Request Message
SymbolIndex	4	4	Binary	The ID (from the Symbol Index msg/Outright Series Index msg/Complex Series Index msg) of the symbol for which a refresh is requested. To request a refresh for all symbols in the channel, set this field to 0.
SourceID	8	10	ASCII	The ID of the client requesting this retransmission. All trailing characters should be NULL. Examples: • 10 character: ABCDEFGHIJ • 9 character: ABCDEFGHI<NULL>
ProductID	18	1	Binary	The unique ID of the feed for which the refresh is requested (listed in the feed's client specification).
ChannelID	19	1	Binary	The ID of the multicast channel for which the refresh is requested.

7.4 SYMBOL INDEX MAPPING REQUEST MESSAGE (MSG TYPE 13)

This message is sent by clients via TCP/IP requesting the Symbol Index Mapping messages for one or all symbols in a specified channel.

The Symbol Index Mapping Request messages should be sent in a packet whose Packet Header Delivery Flag is set to 11, and which contains a valid sequence number.

FIELD NAME	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 21 Bytes
MsgType	2	2	Binary	The type of this message: 13 – Symbol Index Mapping Request Message
SymbolIndex	4	4	Binary	The ID (from the Symbol Index msg/Outright Series Index msg/Complex Series Index msg) of the symbol for which a refresh is requested. To request a refresh for all symbols in the specified channel, set this field to 0.
SourceID	8	10	ASCII	The ID of the client requesting this retransmission. All trailing characters should be NULL. Examples: • 10 character: ABCDEFGHIJ • 9 character: ABCDEFGHI<NULL>
ProductID	18	1	Binary	The unique ID of the feed for which the refresh is requested (listed in the feed's client specification).
ChannelID	19	1	Binary	The ID of the multicast channel for which the refresh is requested.
RetransmitMethod	20	1	Binary	The delivery method for the requested symbol index mapping information. Valid values: ▪ 0 – deliver via UDP

8. Pillar Request Server - Response Messages

8.1 REQUEST RESPONSE MESSAGE (MSG TYPE 11)

This message will be sent immediately via TCP/IP in response to the client's request for retransmission, refresh or Symbol Mapping messages.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 29 Bytes
MsgType	2	2	Binary	The type of this message: 11 – Request Response Message
RequestSeqNum	4	4	Binary	The sequence number of the request message sent by the client. This can be used by the client to couple this response with the original request message.
BeginSeqNum	8	4	Binary	For Retrans Request responses, the beginning sequence number of the requested retransmission range. For responses to Refresh or Symbol Mapping Requests, 0.
EndSeqNum	12	4	Binary	For Retrans Request responses, the ending sequence number of the requested retransmission range. For responses to Refresh or Symbol Mapping Requests, 0.
SourceID	16	10	ASCII	The ID of the client requesting this retransmission. All trailing characters should be NULL.
ProductID	26	1	Binary	The unique ID of the feed for which the request was made (listed in the feed's client specification).
ChannelID	27	1	Binary	The ID of the multicast channel for which the request was made.
Status	28	1	ASCII	The reason why the request was rejected. Valid values: ▪ 0 – Message was accepted ▪ 1 – Rejected due to an Invalid Source ID ▪ 3 – Rejected due to maximum sequence range (see threshold limits) ▪ 4 – Rejected due to maximum request in a day ▪ 5 – Rejected due to maximum number of refresh requests in a day ▪ 6 – Rejected. Request message SeqNum TTL (Time to live) is too old. Use refresh to recover current state if necessary. ▪ 7 – Rejected due to an Invalid Channel ID ▪ 8 – Rejected due to an Invalid Product ID ▪ 9 – Rejected due to: 1) Invalid MsgType, or 2) Mismatch between MsgType and MsgSize

8.2 HEARTBEAT RESPONSE MESSAGE (MSG TYPE 12)

Clients who remain connected to the Retransmission Server intraday must respond to a Heartbeat with a Heartbeat Response message within 5 seconds. If no timely client response is received, the connection will be closed.

FIELD NAME	OFFSET	SIZE	FORMAT	DESCRIPTION
MsgSize	0	2	Binary	Size of the message: 14 Bytes
MsgType	2	2	Binary	The type of this message: 12 – Heartbeat Response message
SourceID	4	10	ASCII	The ID of the client requesting this retransmission . All trailing characters should be NULL.

9. Operational Information

9.1 SYSTEM BEHAVIOR ON START AND RESTART

At system startup or at start of system recovery following a failure, PILLAR feeds send the following messages over each channel:

1. Multicast priming from the primary Publisher's source IPs: a series of Heartbeats OR Sequence-Reset messages for several seconds and sequence number set to 1. (Packet Delivery Flag is set to 1 for heartbeats and 12 for Sequence-Reset messages)
2. Sequence Number Reset Message (Msg Type 1), the sequence number is 1 and the packet DeliveryFlag is 12
3. For securities published on this channel, a full spin of:
    a. Symbol Index Mapping Messages (Msg Type 3) / Outright Series Index Mapping Message (Msg Type 50) / Complex Series Index Mapping Message (Msg Type 60)
    b. Symbol Clear Message (Msg Type 32)
    c. Security Status Message (Msg Type 34) / Options Status Message (Msg Type 51)

9.2 PILLAR PUBLISHER FAILOVER

When failing over to the backup PILLAR Publisher, the following refresh information is published.

Note: During the failover refresh, DeliveryFlag fields for all Packet Headers except Heartbeats are set to 10.

1. Multicast priming from the backup Publisher's source IPs: a series of Heartbeats OR Sequence-Reset messages for several seconds with the sequence number set to 1. (Packet Delivery Flag is set to 1 for heartbeats and 12 for Sequence-Reset messages).
2. A Sequence Number Reset Message (Msg Type 1) is sent in its own packet
3. For each symbol or series, the following are published,
    - Symbol Index Mapping Messages (Msg Type 3) / Outright Series Index Mapping Message (Msg Type 50) / Complex Series Index Mapping Message (Msg Type 60)
    - Symbol Clear Message (Msg Type 32)
    - The last Security Status Message (Msg Type 34) / Options Status Message (Msg Type 51)
    - All required refresh messages
    - The last Source Time Reference Message (Msg Type 2)

Once all symbols or series have been refreshed, Packet Header DeliveryFlag fields return to the normal 11.

9.3 DISASTER RECOVERY SITE

All PILLAR feeds are published out of the NYSE Mahwah data center. In case of catastrophic failure in Mahwah, all affected systems including PILLAR feeds will be coldstarted at the Cermak Disaster Recovery site in Chicago. The Cermak configuration of channels and multicast groups is identical to Mahwah for all feeds, except the source IPs are different. The initial publication sequence is as described in System Behavior on Start and Restart.

9.4 PROPRIETARY DATA PRODUCTION HOURS (US EASTERN TIME)

EVENT	NYSE TAPE A	NYSE TAPES B&C	ARCA	AMERICAN NATIONAL CHICAGO	ARCA OPTIONS AMERICAN OPTIONS
Sequence Number Reset	12:15am*	12:15am*	12:15am*	12:15am*	12:15am*
Symbol Mapping	12:15am*	12:15am*	12:15am*	12:15am*	12:15am*
Early Open Auction		7:00am	4:00am	7:00am	
Core Open Auction and Open	9:30am	9:30am	9:30am	9:30am	9:30am
Closing Auction and Close	4:00pm	4:00pm	4:00pm	4:00pm	4:00pm/4:15pm **
End of Late Session			8:00pm	8:00pm	

*System startup time. Actual message publication will not occur prior to this time, and will commence shortly afterward, when startup is completed.

** Please refer to market hours on https://www.nyse.com/markets/hours-calendars

.

9.5 NYSE PILLAR CERT TESTING

NYSE Pillar CERT connection information is available on the public NYSE Proprietary IP Address spreadsheet.

CERT testing hours are approximately 1:00 AM until 8:15 PM.

- Email: Technology Member Services: tms@nyse.com
- Telephone: +1 212 896-2830 x2

9.5.1 Pillar Request Server Certification

Customers of the new Pillar Request Server for Refresh/Retransmission functionality must certify their readiness in the NYSE Pillar CERT environment before sessions are available in production.

The source IPs of the customer sessions are required as part of the production Pillar Request Server setup.

10. Symbol Index Mapping File

For customers that prefer to download the symbol index mapping from an FTP server, a file containing the symbol index mapping information is made available via FTP every trading day

10.1 SYMBOL INDEX MAPPING FILE FORMAT

FIELD NAME	FORMAT	DESCRIPTION
Symbol	ASCII	The full symbol in NYSE Symbology.
CQS Symbol	ASCII	The full symbol in CTS and UTP line format. See NYSE Symbology.
SymbolIndex	Numeric	The unique ID for this symbol. See Symbol Indexes. This ID is unique for products within each market. It cannot be used to cross reference a security between markets.
NYSE Market	Character	The market within the NYSE Group where this symbol is traded: • N – NYSE • P – NYSE Arca • C – NYSE National • A – NYSE American • M - NYSE Chicago
Listed Market	Character	For listed equity markets, the market where this symbol is listed: • A – NYSE American • N – NYSE • L - LTSE • P – NYSE Arca • Q – NASDAQ • V - IEX • Z – CBOE
TickerDesignation	Character	The SIP tape on which this symbol is published: • A – CTA Tape A • B – CTA Tape B • C – CTA Tape C
UOT	Numeric	The number of shares in a round lot.
PriceScaleCode	Numeric	A code used to place the decimal point in all price fields for this symbol. See section 3.5 for details on price handling.
SystemID	Numeric	The ID of the matching engine instance that handles this symbol.
Bloomberg BSID	Empty	Reserved field. No character between the delimiting pipe characters.
Bloomberg Global ID	Empty	Reserved field. No character between the delimiting pipe characters.

NOTE: The pipe delmited (.txt) version of this file has an additional pipe character at the end of the Bloomberg Global ID field, so every record in the file has three pipe characters after the System ID field.

10.2 SYMBOL INDEX MAPPING FILE

The symbol index mapping files are available by 11pm EST at the following public locations:

- **NYSE**

Pipe-delimited: ftp://ftp.nyse.com/NYSESymbolMapping/NYSESymbolMapping.txt
XML format: ftp://ftp.nyse.com/NYSESymbolMapping/NYSESymbolMapping.xml

- **NYSE American**

Pipe-delimited: ftp://ftp.nyse.com/AmericanSymbolMapping/AmericanSymbolMapping.txt
XML format: ftp://ftp.nyse.com/AmericanSymbolMapping/AmericanSymbolMapping.XML

- **NYSE Arca**

Pipe-delimited: ftp://ftp.nyse.com/ARCASymbolMapping/ARCASymbolMapping.txt
XML format: ftp://ftp.nyse.com/ARCASymbolMapping/ARCASymbolMapping.XML

- **NYSE Chicago**

Pipe-delimited: ftp://ftp.nyse.com/ChicagoSymbolMapping/ChicagoSymbolMapping.txt
XML format: ftp://ftp.nyse.com/ChicagoSymbolMapping/ChicagoSymbolMapping.XML

- **NYSE National**

Pipe-delimited: ftp://ftp.nyse.com/NationalSymbolMapping/NationalSymbolMapping.txt
XML format: ftp://ftp.nyse.com/NationalSymbolMapping/NationalSymbolMapping.XML

11. Options Index Mapping File

For customers that prefer to download the symbol/series index mapping from an FTP server, a file containing the symbol/series index mapping information is made available via FTP every trading day by 11pm EST. The download file is not updated intraday to keep up with these changes.

11.1 OPTIONS INDEX MAPPING FILE

FTP files are available every trading day at:

ftp://ftp.nyse.com/ArcaAmexOptionsSymbolMapping/

This directory contains six files, for:

American production, American certification environment (legacy XDP*)

American production, American certification environment (PILLAR)

Arca production, Arca certification environment (legacy XDP*)

Arca production, Arca certification environment (PILLAR)

*__Note__ legacy XDP symbol mapping file will be decomissioned once symbol migration to Pillar is fully completed. ARCA scheduled for Q4 ’21 and American in Q1 ‘22

The options index mapping files are available by 11pm EST at the following public locations:

- **NYSE American**

Legacy XDP

Pipe-delimited

production: ftp://ftp.nyse.com/AmericanSymbolMapping/AmexOptionsSymbolMapping.txt

certification: ftp://ftp.nyse.com/AmericanSymbolMapping/AmexOptionsSymbolMapping_cert.txt

PILLAR

Pipe-delimited

production: ftp://ftp.nyse.com/AmericanSymbolMapping/PILLAR_AmexOptionsSymbolMapping.txt

certification: ftp://ftp.nyse.com/AmericanSymbolMapping/PILLAR_AmexOptionsSymbolMapping_cert.txt

- **NYSE Arca**

Legacy XDP

Pipe-delimited

production: ftp://ftp.nyse.com/ArcaOptionsSymbolMapping/ArcaOptionsSymbolMapping.txt

certification: ftp://ftp.nyse.com/ArcaSymbolMapping/ArcaOptionsSymbolMapping_cert.txt

PILLAR

Pipe-delimited

production: ftp://ftp.nyse.com/ArcaOptionsSymbolMapping/PILLAR_ArcaOptionsSymbolMapping.txt

certification: ftp://ftp.nyse.com/ArcaSymbolMapping/PILLAR_ArcaOptionsSymbolMapping_cert.txt

11.2 OPTIONS INDEX MAPPING FILE FORMAT (LEGACY XDP)

All FTP files are in the same pipe-delimited format. The file contains 3 different row types, corresponding to the 3 types of mapping messages:

- Symbol mapping MsgType field = 435
- Series mapping MsgType field = 437
- Complex series mapping MsgType field = 439

11.2.1 Underlying symbol mapping record format (MsgType 435)

MsgType|UnderlyingIndex|UnderlyingSymbol|MarketID|SystemID|ExchangeCode|PriceScaleCode|SecurityType|PriceResolution| TopFeedChannelID|DeepFeedChannelID|ComplexFeedChannelID

Sample

435|2872|YANG|4|14|P|4|E|5|31|67|121 435|2873|YCS|4|14|P|4|E|5|31|67|121 435|2874|YELP|4|14|N|4|C|5|31|67|121

11.2.2 Series mapping record format (MsgType 437)

MsgType|StreamID|SeriesIndex|MarketID|SystemID|UnderlyingIndex|ContractMultiplier|MaturityDate|PutOrCall|StrikePrice|PriceScaleCode| UnderlyingSymbol|OptionSymbolRoot|GroupID

Sample

437|225|31717725|4|14|2872|100|160115|1|30|4|YANG|YANG |143601
437|225|31717726|4|14|2872|100|160115|1|35|4|YANG|YANG |143601
437|225|31717727|4|14|2872|100|160115|1|40|4|YANG|YANG |143604

11.2.3 Complex series mapping record format (MsgType 439)

MsgType|StreamID|ComplexIndex|ComplexSymbol|MarketID|SystemID|NoOfLegs|SymbolIndex|LegRatioQty|Side|SecurityType

The last four fields in red contain information for a single leg and repeat as many times as the number of legs.

Sample

439|227|31731777|4YOKU15289247|4|14|2|31722253|1|S|O|31722254|1|B|O
439|227|31731778|4YOKU15289373|4|14|2|31722643|1|S|O|31722644|1|B|O
439|227|31731779|4YHOO153041|4|14|2|31720592|1|S|O|31721006|1|S|O

11.3 OPTIONS INDEX MAPPING FILE FORMAT (PILLAR)

All FTP files are in the same pipe-delimited format. The file contains 3 different row types, corresponding to the 3 types of mapping messages:

- Symbol mapping MsgType field = 3
- Series mapping MsgType field = 50
- Complex series mapping MsgType field = 60

11.3.1 Underlying symbol mapping record format (MsgType 3)

MsgType|UnderlyingIndex|UnderlyingSymbol|MarketID|SystemID|ExchangeCode|PriceScaleCode|SecurityType|PriceResolution|TopFeedChannelID|DeepFeedChannelID|ComplexFeedChannelID

11.3.2 Series mapping record format (MsgType 50)

MsgType|SeriesIndex|MarketID|SystemID|UnderlyingIndex|ContractMultiplier|MaturityDate|PutOrCall|StrikePrice|PriceScaleCode|UnderlyingSymbol|OptionSymbolRoot|Reserved|SeriesType|ClosingOnlyIndicator

11.3.3 Complex series mapping record format (MsgType 60)

MsgType|ComplexIndex|MarketID|SystemID|NoOfLegs|SymbolIndex|LegRatioQty|Side|SecurityType

The last four fields in red contain information for a single leg and repeat as many times as the number of legs.