Document title

NYSE ARCABOOK FOR OPTIONS CLIENT SPECIFICATION

FOR NYSE ARCA OPTIONS AND NYSE AMEX OPTIONS EXCHANGES

Version	Date
3.3	5 Mar 2014

PREFACE

DOCUMENT HISTORY

The following table provides a description of the current version change history. For historical document history details please see Appendix A.

VERSION NO.	DATE	CHANGE DESCRIPTION
3.15	27 Sep 2012	■ This version of the Client Specification is an amalgamation of the *NYSE ArcaBook for Options Client Specification*, the *NYSE ArcaBook for Complex Options Client Specification* and the *NYSE ArcaTrade for Options Client Specification*. ■ Numerous modifications have been made to the Specification to make it more useful to the end user. ■ The document has been rebranded with the new NYSE Technologies template.
3.16	11 Mar 2013	■ Updated 4.2.4 and 4.3.3 Auction ID replaced by Price and Side added.
3.17	08 May 2013	■ Additional information to section 4.3.3 Complex RFQ Message
3.18	28 Jan 2014	■ Section 5.2.5 Table 43 Pad field removed and Message Change Bytes amended ■ Section 5.1.6 Table 38 replaced by table 22
3.19	03 Feb 2014	■ Section 4.3.5 added new event code in Event Code field 'O' – Open Series
3.2	13 Feb 2014	■ Section 4.3.5 additional event code changes
3.3	5 Mar 2014	■ Section 4.3.2 updated Complex Option Leg definition to accommodate complex symbol release ■ Section 4.3.2, Table 22 updated to include symbol field ■ Section 5.1.6, Table 38 updated to include symbol field

CONTACT INFORMATION

Service Desk

- Telephone: +1 212 383 3640 (International)
- Telephone: 866 873 7422 (Toll free, US only)
- Email: service.desk@nyx.com

FURTHER INFORMATION

- For additional product information, visit:
  - NYSE Arca Options: http://www.nyxdata.com/page/1170
  - NYSE Amex Options: http://www.nyxdata.com/page/1165

- For updated capacity figures, visit our capacity pages at: http://www.nyxdata.com/capacity
- For details of IP addresses, visit our IP address pages at: http://www.nyxdata.com/ipaddresses
- For a full glossary, visit: http://www.nyxdata.com/glossary/

CONTENTS

1. INTRODUCTION

NYSE ArcaBook for Options is a real-time data feed that disseminates option order/consolidated book and trades/last sale information to subscribers for NYSE Arca exchanges. NYSE ArcaBook allows subscribers to produce and display the NYSE Arca open order or consolidated book and ticker. Order routing algorithms can also use NYSE ArcaBook data.

1.1 DEPTH OF BOOK MESSAGING

NYSE Arca is implementing an innovative approach with its Depth of Book Quotation feed to help subscribers implement fast algorithms to update their top five quotes. Each Quote Message has two price levels identified by an index (1-5):

- **Delete Price Level** The index of a price level that is no longer in the top five. This entire price level should be deleted before inserting a new price level.

  If this index is the same as the Insert price level index, this is an update message for this price level.

- **Insert Price Level** The index of a new price level to be inserted. If this index is the same as the Delete price level, this message simply updates the current price or volume of this price level.

Messages for price changes that re-sequence the top five price levels in the Depth of Book will include one price level to delete and a different price level to insert.

Messages for volume-only changes or changes in price that do not re-sequence other price levels will have the same index for both the Delete and Insert price level. See Depth of Book Messaging for examples.

1.2 NYSE ARCABOOK FOR OPTIONS INTERFACE

This Specification is for developers that wish to write applications that interface with the following NYSE ArcaBook data feeds:

INTERFACE	DESCRIPTION
Options	A FAST compacted data feed for aggregate option quotes of the consolidated book.
Complex Options	A binary data feed for complex options top of book quotes.
Trades	A data feed for option trades and trade modifications. Trade data only reflects trades that take place within NYSE Arca trading platforms, rounded down to the nearest lot. Routed, mixed, and odd lot trades may be added to the feed at a later date.

The interface is message-based, using fixed length messages over a UDP IP Multicast with binary number and fixed length ASCII fields. Binary values are in network order (Big-Endian) format.

The interface contains the following categories of messages:

- UDP (User Datagram Protocol) Multicast, to broadcast trade details
- TCP/IP Recovery, to recover packets dropped by the UDP multicast

For details of multicast channels and IP addresses, see the 'Amex Options' and 'Arca Options' tabs on http://www.nyxdata.com/ipaddresses.

1.2.1 Options Data Feeds

Customers can subscribe to the following data feeds:

- **Top of Book** The single best price level in the Consolidated Book; this has a price-level index of 1.
- **Depth of Book** The top five price levels in the Consolidated Book; these have price-level indices of 1 (top of the book) through 5.
- **Options Series Status** Index mapping messages are sent as initial downloads in all channels.
- **Options Order Imbalance** Sent in response to orders submitted during pending auctions.

The following messages are available on the Options feeds:

- **Underlying Index Mapping** For details, see Underlying Index Mapping
- **Series Index Mapping** For details, see Series Index Mapping
- **OCC Symbology Update** For details, see OCC Symbology Update
- **Aggregate Quote Message** For details, see Aggregate Quote Message
- **Imbalance Message** For details, see Imbalance Message
- **Crossing RFQ Message** For details, see Crossing RFQ Message
- **System Event Message** For details, see System Event Message

1.2.2 Complex Options Data Feed

Customers must connect to one multicast group and make one TCP/IP connection in order to received the Complex Leg Definitions and to process Top of Book Quotes for complex orders.

The following messages are available on the NYSE ArcaBook for Options feeds:

- **Underlying Index Mapping** For details, see Underlying Index Mapping
- **Series Index Mapping** For details, see Series Index Mapping
- **OCC Symbology Update** For details, see OCC Symbology Update
- **Complex Option Leg Definition Message** Complex Leg definitions indicate the available legs submitted in a complex order. Each definition will have a complex. For details, see Complex Option Leg Definition Message
- **Complex RFQ Message** These messages are sent out during complex order auction. For details, see Complex RFQ Message
- **Top Quote Message** The single best price-level in the Consolidated Book. This has a price-level index of 1. Quotes are referenced by the Complex ID. For details, see Top Quote Message
- **System Event Message** For details, see System Event Message

1.2.3 Trades Data Feed

The Trades data feed includes the following messages:

- **Underlying Index Mapping** For details, see Underlying Index Mapping

- **Series Index Mapping** For details, see Series Index Mapping
- **OCC Symbology Update** For details, see OCC Symbology Update
- **Last Sale Message** For details, see Last Sale Message
- **Trade Bust or Correction Message** For details, see Trade Bust or Correction Message
- **System Event Message** For details, see System Event Message

1.3 NYSE ARCA API CERTIFICATION

Subscribers must certify their NYSE ArcaBook subscription clients with NYSE Arca. NYSE Arca provides an IP address, port number, username and password to use for testing. To schedule a test, please call the **Service Desk**.

1.4 COMPACTION

This section applies only to FixFast feed, not to binary feed. Trades are not compacted.

NYSE Arca compacts all NYSE ArcaBook for Options messages using the FIX Adapted for Streaming (FAST) Protocol. See FIX FAST Protocol for a detailed explanation and encoding methods.

Compacted messages are a header and body with no delimiters between messages, as shown below:

MESSAGE HEADER	MESSAGE HEADER	MESSAGE HEADER	MESSAGE HEADER	...
Body	Body	Body	Body	...

2. COMMUNICATION

2.1 ACCESS

NYSE ArcaBook for Options subscribers connect to multicast addresses for the primary data feeds and can also connect to a TCP/IP server for packet retransmissions.

For Complex Options there are two types of complex feeds:

- FixFast is the existing complex feed where one gets leg definitions and top of the book quotes in compact format.
- Binary feed is the new format where one gets leg definitions, top of the book quotes and RFQs for complex auctions. The feed will be in uncompacted format. This feed is also referred to as Complex Consolidated feed.

All Trades data will be disseminated from approximately 9:30am to 4:00pm EST. The feed may be accessible for connectivity prior to 3:30am EST depending on start-of-day processing.

2.2 MULTICAST FEEDS

Multicasts for NYSE ArcaBook for Options use UDP (User Datagram Protocol).

Each subscription address has two data feeds each with an IP address and port. For details of IP addresses, visit our IP address pages at: http://www.nyxdata.com/ipaddresses.

In the event that a packet is lost on the primary feed for a subscription, clients can retrieve the lost packet from the secondary feed. Because UDP is unreliable and may drop packets from both feeds, a TCP/IP Recovery Server is provided from which clients can request dropped packets. See TCP/IP Recovery Messages for more information.

2.2.1 NYSE ArcaBook for Options Multicast Quote Feeds

Note: The number of subscriptions and their configuration has not been finalized. This information is subject to change.

Data feeds for specific underlying stocks are sent to different multicast subscription addresses. Top of Book uses a separate subscription address from Depth of Book. This addressing scheme allows customers to subscribe to the specific data feeds they need. The Penny Pilot Issues have been separated on to three separate subscriptions.

NYSE ArcaBook for Options Quote Messages are compacted before transmission and several are transmitted in a single packet. Each packet has a header containing the packet size and sequence number. Packet headers are not compacted. Clients expand compacted Quote messages before processing them.

2.2.1.1 NYSE ArcaBook for Options AB Consolidated Channel

This channel will send all the series status messages as well as crossing RFQs for the regular series. This channel will be uncompacted and will replace the existing symbol status channel. There are two differences between this channel (subscription 122) and the existing series status channel (subscription 125):

- Channel 122 has crossing RFQ messages in addition to series status messages
- Channel 122 is uncompacted whereas channel 125 is compacted

2.2.2 Trades Feed

Note: All Trades data will be disseminated to a single multicast address. The number of subscriptions and their configuration is subject to change. Trade messages are sent in packets. Several messages can be transmitted in a single packet. Each packet has a header containing the packet size and a packet sequence number.

2.2.3 TCP/IP Recovery

Subscribers may connect to the TCP/IP Recovery Server to request dropped packets from the multicast feed. The Recovery Server accepts connections on predefined addresses and ports and requires a login before responding to requests. It accepts primary and backup connections to assist recovery on the subscriber's end.

NYSE Arca supplies subscribers with the following parameters:

- An IP address
- A port
- A username
- A password

Subscribers supply NYSE Arca with the IP address for their connection.

Please note that in order to utilize either TCP/IP Recovery Server in either Test (Cert) or Production , you will need a SourceID and Password assigned to you. Please contact the Service Desk to request SourceID(s) information.

2.2.4 Bandwidth Requirements

The recommended minimum bandwidth for multicast is 1.5 Megabits per second, roughly a T1 connection.

NYSE Arca offers connectivity to both its Chicago and New Jersey data centers. Clients are strongly recommended to implement redundant connectivity to ensure they continue to receive Trades data in the event of issues with their primary connection.

For Complex Options traffic the following message rates and sizes are suggested:

Table 1 Complex Options Bandwidth

RATE/SIZE	ESTIMATE
Complex Options bandwidth (Mbps)	5
Peak message per second rate*	2,000
Packet size	Typically up to 100 bytes**
Maximum number of packets in a day	TBD
Maximum total number of individual quotes messages in a day	TBD

* Function of number of Complex Option instruments; estimate of 2,000 messages per second as peak rate assumes 2,000 Complex Option instruments retained overnight and initial mapping for which to be sent out at start of day.

** Packet size is variable; usually packet will be one message.

3. MESSAGES

3.1 DATA TYPES

All numeric fields are in unsigned binary. Binary data is in network order (Big-Endian) format. All alphanumeric fields are left-justified and null-padded.

3.1.1 Sequence Numbers

Sequence Numbers for packets and for messages are four-byte integers. These numbers start the data feed session at 1 and increment by 1 for each new packet or message. See NYSE ArcaBook for Options Recovery Server Messages for more information on sequence numbers.

3.1.2 Prices

Prices are four-byte integers in binary scaled to four decimal positions. To determine the decimal price, divide the whole integer price by 10,000.

- Example 1: Whole integer price is 135000. The decimal price is 135000 ÷ 10,000 = 13.50
- Example 2: Whole integer price is 13500. The decimal price is 13500 ÷ 10,000 = 1.35

3.1.3 Complex ID and Trade Reference ID

The Complex ID and Trade Reference ID will be a value of a 64 bit long. To convert it in a 32 bit processing environment the following example can be used. **Note that compaction will use the structure below**.

Table 2 Example: Big-Endian Environment

	COMPLEX ID				TRADE REF. ID			NOTES AND VALUES
	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	OFFSET	SIZE (BYTES)	FORMAT	
ID	0	4	Binary	AB_ORDER_ID	0	4	Binary	OrderID (1 – 4,294,967,294)
MarketID	4	2	Binary	AB_MARKET_ID	4	2	Binary	Market ID (0-65,536)
SystemID	6	1	Binary	AB_SYSTEM_ID	6	1	Binary	Matching engine number (0-256)
Bit	7	1	Char	AB_BIT	-	-	-	NULL
Pad	-	-	-	-	7	1	-	-

Table 3 Example: Little-Endian Environment

	COMPLEX ID				TRADE REF. ID			NOTES AND VALUES
	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	OFFSET	SIZE (BYTES)	FORMAT	
Pad	-	-	-	-	0	1	-	-
Bit	0	1	Char	AB_BIT	-	-	-	NULL
SystemID	1	1	Binary	AB_SYSTEM_ID	1	1	Binary	Matching engine number (0-256)
MarketID	2	2	Binary	AB_MARKET_ID	2	2	Binary	Market ID (0-65,536)
ID	4	4	Binary	AB_ORDER_ID	4	4	Binary	OrderID (1 – 4,294,967,294)

3.1.4 Timestamps

The timestamp field is a four-byte integer that provides time in milliseconds starting from midnight (00:00:00:000) of the trading day. NYSE ArcaBook for Options computes timestamps as: seconds x 1000 + milliseconds. For example, the timestamp for 10:00:00:376 is converted to:

(36000 x 1000) + 376 = 36000376.

3.2 PACKETS

All packets are encapsulated in variable length Transmission Blocks, as shown in **Table 4**.

Table 4 Packet Transmission Block

PACKET LENGTH	TYPE	SUBSCRIPTION	PACKET SEQUENCE NUMBER	COMPACTED MESSAGES
The full length of the packet as two-byte Numeric Binary	A one-byte Alpha/Numeric code: ■ 'M' - Message ■ 'B' - Heartbeat ■ 'N' - Not found (for TCP/IP packet replay only)	The subscription number for the packet as one-byte Numeric Binary (0 -255)	A four-byte Numeric Binary. For heartbeat packets (Type=B), this is the last Packet Sequence Number sent.	Messages are not present in heartbeat or not found packets (Type B or N).

Compacted messages within a packet may be for different option series.

Heartbeat packets contain no compacted messages and do not increment the Packet Sequence Number. Heartbeat packets are only sent during periods of inactivity to indicate the connection is still open. They are sent for both FixFast and Binary feeds.

3.3 RECOVERY

In the event that a packet is lost on the primary feed for a subscription, clients can check the secondary feed for the lost packet. If both feeds have dropped the packet, clients can request retransmission from the TCP/IP Recovery Server.

Clients use the subscription number and the packet sequence number to request missing packets in the Dropped Packet Request Message. Packet sequence numbers start from one each day for a specific multicast subscription number.

Clients can continue to process subsequent packets even though a packet is missing. Each message within a packet also has a message sequence number. Message sequence numbers also start from one each day and are incremented by option series or by one per unique Complex Option ID. When packets are dropped, clients can still process messages in subsequent packets for any series that is not missing messages based on the message sequence numbers.

4. UDP MULTICAST MESSAGES

4.1 COMMON MESSAGES (OPTIONS, COMPLEX OPTIONS & TRADES)

4.1.1 OCC Symbology Update

The official 21-character symbology key will map exactly to the fields underlined in green in the *Series Index Mapping* below.

Table 5 New OCC Explicit Series Key

SYMBOL	YR	MO	DAY	C/P	EXPLICIT STRIKE	DECIMAL
MSFT	06	03	18	C	00047	500

For symbols that are under symbology consolidation and multiple trading option classes the fields will be:

- For trading class 1
  - UnderlyingIndexMappingMsg.Symbol=MSFT
  - SeriesIndexMappingMsg.UnderlyingSymbol=MSFT1
- For trading class 2
  - UnderlyingIndexMappingMsg.Symbol=MSFT
  - SeriesIndexMappingMsg.UnderlyingSymbol=MSFT2
- And so forth.

In all cases, once OCC symbology goes live, option symbol in the Series Index Mapping message will be NULL.

4.1.2 Series Index Mapping

Series Index messages will immediately follow the underlying index messages. Each subscription will send the series mappings used on its subscription.

Table 6 Series Index Mapping Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
Header (8 bytes)					
Message Length	0	2	Binary		60
Message Type	2	1	Alpha/Numeric	AQ_MSG_TYPE	‘m’
Subscription	3	1	Binary	AQ_SUBSCRIPTION	0-255 identifying the Subscription
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since Midnight
Message Body (52 Bytes)					

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
Series Index	8	4	Binary	AQ_SERIES_INDEX	Series Mapping Index
Market ID	12	2	Binary	AQ_MARKET_ID	Identifies Market Origin: 0 - 65,535
System ID	14	1	Binary	AQ_SYSTEM_ID	Identifies Trading Engine: 0 - 255
Bit	15	1	Alpha	AQ_BIT	NULL
Underlying Index	16	4	Binary	AQ_UNDERLYING_INDEX	Underlying Stock Mapping Index
Reserved1	20	2	Binary	AQ_RESERVE_1	NULL
Reserved2	22	1	Binary	AQ_RESERVE_2	NULL
Reserved3	23	1	Alpha	AQ_RESERVE_3	NULL
Reserved6	24	4	Binary	AQ_RESERVE_6	NULL
Underlying Quantity	28	4	Binary	AQ_UNDERLYING_QUANTITY	0. Reserved for Future Use.
Underlying Symbol	32	6	Alpha	AQ_SYMBOL	Underlying Stock Ticker Symbol – (Left Justified - space padded)
Expiry Year	38	2	Alpha	AQ_EXPIRE_YEAR	YY
Expiry Month	40	2	Alpha	AQ_EXPIRE_MONTH	MM
Expiry Day	42	2	Alpha	AQ_EXPIRE_DAY	DD
Put or Call	44	1	Alpha	AQ_PUT_CALL	■ 'P' - Put ■ 'C' - Call
Strike Price Whole	45	5	Alpha	AQ_STRIKE_PRICE	Right Justified, Zero Filled
Strike Price Decimal	50	3	Alpha	AQ_STRIKE_DECIMAL	Left Justified, Zero Filled
Price Scale	53	1	Binary	AQ_PRICE_SCALE	Decimal places on price
Option Symbol	54	5	Alpha	AQ_OCC_SYMBOL	The full OPRA symbol for this series. Left Justified - NULL padded.
Reserved4	59	1	Alpha	AQ_RESERVE_4	NULL

4.1.3 Underlying Index Mapping

Underlying Index Messages will be the first messages sent down the multicast feeds. Each subscription will send the underlying mappings used on its subscription.

Table 7 Underlying Index Mapping Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
Header (8 bytes)					
Message Length	0	2	Binary		32
Message Type	2	1	Alpha/Numeric	AQ_MSG_TYPE	‘n’
Subscription	3	1	Binary	AQ_SUBSCRIPTION	0-255 identifying the Subscription
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since Midnight
Message Body (24 Bytes)					
Underlying Index	8	4	Binary	AQ_UNDERLYING_INDEX	Underlying Stock Mapping Index
Market ID	12	2	Binary	AQ_MARKET_ID	Identifies Market Origin: 0 - 65,535
System ID	14	1	Binary	AQ_SYSTEM_ID	Identifies Trading Engine: 0 - 255
Bit	15	1	Alpha	AQ_BIT	NULL
Reserved6	16	4	Binary	AQ_RESERVE_6	NULL
Price Scale	20	1	Binary	AQ_PRICE_SCALE	Decimal places on price
Price Resolution	21	1	Alpha	AQ_PRICE_RESOLUTION	■ ‘0’ - Penny ■ ‘1’ - Penny/Nickel ■ ‘5’ - Nickel/Dime
Exchange Code	22	1	Alpha	AQ_EXCHANGE_CODE	■ ‘N’ - NYSE ■ ‘Q’ - Nasdaq ■ ‘P’ - Arca ■ ‘A’ - Amex
Security Type	23	1	Alpha	AQ_SECURITY_TYPE	■ ‘A’ - ADR ■ ‘C’ - Common ■ ‘E’ - ETF ■ ‘F’ - Foreign

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
					■ ‘I’ - Units ■ ‘M’ - Misc. ■ ‘P’ - Preferred ■ ‘R’ - Rights ■ ‘S’ - Shares of Ben Int ■ ‘T’ - Test ■ ‘U’ - Units ■ ‘W’ - Warrant
Symbol	24	6	Alpha	AQ_SYMBOL	Stock Ticker Symbol, Left Justified - NULL Padded
Reserved3	30	1	Alpha	AQ_RESERVE_3	NULL
Reserved4	31	1	Alpha	AQ_RESERVE_4	NULL

4.2 OPTIONS

4.2.1 Header Format

All NYSE ArcaBook for Options messages are sent over multicast. Once uncompacted, they all begin with the following header.

Table 8 Header Format

HEADER	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	0 – 65535 (value includes 8 byte header)
Message Type	2	1	Alpha/Numeric	A single character to identify the message: ■ ‘q’ - quote ■ ‘i’ - auction imbalance ■ ‘v’ - system event
Subscription	3	1	Binary	0-255 identifying the subscription.
Time Stamp	4	4	Binary	Milliseconds since midnight

4.2.2 Aggregate Quote Message

Top-of-Book and Depth of Book Quote messages use the same data structure that identifies one price level for one side of the NYSE ArcaBook for Options Book. Each price level has an index number (1-5) indicating its placement in the book from best to worst (within the levels of the Depth of Book data feed).

Each Quote message has two price level fields identifying the price-level index and the operation that the message indicates for that level:

- Delete price level = the index of the price level that should be deleted
- Insert price level = the index of the price level that should be inserted

The Depth of Book quote feed uses these two price-level fields to maintain the five price levels in this feed. The Top of Book quote feed always modifies price level 1 (for the top of the book).

The Message Sequence Number is on a per-series/subscriptions basis. Each series has a set of sequence numbers so that if a packet is lost, the book can still be updated without interruption for any series that is not missing messages because of lost packets. Quote Messages and System Event Messages will use the same Message Sequence Numbers because they are on the same subscription. However Auction Imbalance messages will use a different set of Message Sequence Numbers because they are on a different subscription. See NYSE ArcaBook for Options Recovery Server Messages for information on recovering missing packets.

Table 9 Aggregate Quote Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
Header (8 bytes)					
Message Length	0	2	Binary		40
Message Type	2	1	Alpha/Numeric	AQ_MSG_TYPE	‘q’
Subscription	3	1	Binary	AQ_SUBSCRIPTION	0-255 identifying the Subscription
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since Midnight
Message Body (32 Bytes)					
Series Index	8	4	Binary	AQ_SERIES_INDEX	Series Mapping Index
Market ID	12	2	Binary	AQ_MARKET_ID	NULL
System ID	14	1	Binary	AQ_SYSTEM_ID	NULL
Bit	15	1	Alpha	AQ_BIT	NULL
Message Sequence Number	16	4	Binary	AQ_SEQUENCE	ARCA assigned sequence number. 1 – 4,294,967,294 within a series.
Reserved6	20	4	Alpha	AQ_RESERVE_6	NULL
Customer Volume	24	4	Binary	AQ_CUSTOMER_VOLUME	Aggregated customer volume
Total Contracts/Shares	28	4	Binary	AQ_QUANTITY	Aggregated volume of the quote
Price	32	4	Binary	AQ_ PRICE	The price at the insert price level.
Delete Price	36	1	Binary	AQ_DELETE_LEVEL	Price-level index to

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
Level					delete for this quote (1-5).
Insert Price Level	37	1	Binary	AQ_INSERT_LEVEL	New price-level index for this quote (1-5).
Side	38	1	Alpha	AQ_BUY_SELL	■ 'B' - Bid ■ 'S' - Offer
Price Scale	39	1	Binary	AQ_PRICE_SCALE	Null (Not Use)

4.2.2.1 Depth of Book Messaging

In this scenario, the price level indices are the same for insert and delete, indicating that this updates that single price level.

Table 10 Scenario 1: Update to a Single Price Level

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	VOLUME
Bid Quote	3	3	7	500
		PRICE LEVEL	**PRICE**	**VOLUME**
Original Depth of Book		1	10	1000
		2	8	1000
		3	6	200
		4	4	3000
		5	2	400
Modified Depth of Book		1	10	1000
		2	8	1000
		3	7	500
		4	4	3000
		5	2	400

Overwrite Quote

Table 11 Scenario 2: New Price Level Worsens Market

In this scenario, the Insert Price Level is a larger index (i.e., a worse price) than the Delete Price Level. This shifts the price-level index of the prices between the Delete Price Level and the Insert Price Level up.

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	VOLUME
Ask Quote	2	4	9	500
		PRICE LEVEL	PRICE	VOLUME
Original Depth of Book		1	2	1000
		2	4	1000
		3	6	200
		4	8	3000
		5	10	400
Modified Depth of Book		1	2	1000
		2	6	200
		3	8	3000
		4	9	500
		5	10	400

Shift Prices

Insert new price

Table 12 Scenario 3: New Price Level Improves Market

In this scenario, the new price level improves the market and shifts the price levels between the Delete Price Level and the Insert Price Level down.

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	VOLUME
Bid Quote	4	2	9	500
		PRICE LEVEL	**PRICE**	**VOLUME**
Original Depth of Book		1	10	1000
		2	8	1000
		3	6	200
		4	4	3000
		5	2	400
Modified Depth of Book		1	10	1000
		2	9	500
		3	8	1000
		4	6	200
		5	2	400

Shift Prices

Insert new price

Table 13 Scenario 4: Delete on a partial depth book

In this scenario, the depth of the book is less than 5 and a price level must be removed from the book.

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	VOLUME
Bid Quote	2	3	0	0
		PRICE LEVEL	**PRICE**	**VOLUME**
Original Depth of Book		1	10	1000
		2	8	1000
		3	6	200

Shift Prices

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	VOLUME
		4	0	0
		5	0	0
Modified Depth of Book		1	10	1000
		2	6	200
		3	0	0
		4	0	0
		5	0	0

Insert Zeroed Quote

Table 14 Scenario 5: Insert on a partial depth book

In this scenario, the depth of the book is less than 5 and a price level must be inserted into the book.

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	VOLUME
Bid Quote	5	2	8	1000
		PRICE LEVEL	PRICE	VOLUME
Original Depth of Book		1	10	1000
		2	6	200
		3	0	0
		4	0	0
		5	0	0
Modified Depth of Book		1	10	1000
		2	8	1000
		3	6	200
		4	0	0

Shift Prices

Insert New Price

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	VOLUME
		5	0	0

4.2.2.2 Top of Book Messaging

In this scenario, the Top of Book message overwrites the previous quote and improves the market.

Table 15 Scenario 6: Top of Book Improves Market

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	VOLUME
Top Bid Quote	1	1	5	100
		PRICE LEVEL	PRICE	VOLUME
Original Top of Book		1	6	200
Modified Top of Book		1	5	100

Table 16 Scenario 7: Top of Book message clears the quote

In this scenario, the Top of Book message overwrites the previous top of book with a zero volume, zero price quote.

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	VOLUME
Top Bid Quote	1	1	0	0
		PRICE LEVEL	PRICE	VOLUME
Original Top of Book		1	6	200
Modified Top of Book		1	0	0

4.2.3 Imbalance Message

The Imbalance message is sent in response to orders submitted during pending auctions. Imbalance messages are sent to a separate multicast address – they are not part of the Top of Book or Depth of Book feeds.

This message is sent between 5:30am EST and the conclusion of the Opening Auction. NYSE ArcaBook for Options also disseminates imbalance information for Halt Auctions.

4.2.3.1 Market Order Imbalance

The Market Order Imbalance is the imbalance of any remaining Market Orders (or Market-on-Close orders for the Closing Auction) that are not going to be executed in an auction. Calculation of match size and indicative match price remain unchanged.

4.2.3.2 Total Imbalance

The Total Imbalance is the net imbalance of orders at the indicative match price for all orders eligible for the next upcoming Auction. This includes Market (or Market-on-Close) and Limit Orders.

Table 17 Auction Imbalance Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
Header (8 bytes)					
Message Length	0	2	Binary		40
Message Type	2	1	Alpha/Numeric	AQ_MSG_TYPE	‘i’
Subscription	3	1	Binary	AQ_SUBSCRIPTION	0-255 identifying the Subscription
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since Midnight
Message Body (32 Bytes)					
Series Index	8	4	Binary	AQ_SERIES_INDEX	Symbol Mapping Index
Market ID	12	2	Binary	AQ_MARKET_ID	NULL
System ID	14	1	Binary	AQ_SYSTEM_ID	NULL
Bit	15	1	Alpha	AQ_BIT	NULL
Message Sequence Number	16	4	Binary	AQ_SEQUENCE	ARCA assigned sequence number. 1 – 4,294,967,294 within a series.
Contracts/Shares	20	4	Binary	AQ_QUANTITY	Indicative match volume.
Price	24	4	Binary	AQ_ PRICE	The indicative match price
Total Imbalance	28	4	Binary	AQ_TOTAL_IMBALANCE	The total imbalance volume
Market Imbalance	32	4	Binary	AQ_MKT_IMBALANCE	The market imbalance volume
Auction Time	36	2	Binary	AQ_AUCTION_TIME	Projected auction time (hhmm) 0 – 2400
Auction Type	38	1	Alpha	AQ_AUCTION_TYPE	■ ‘O’ - Open

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
					■ 'M' - Market ■ 'H' - Halt ■ 'C' - Closing
Reserved4	39	1		AQ_RESERVE_4	NULL

4.2.4 Crossing RFQ Message

This message is sent out at the beginning of the auction of a cross order, using the AB consolidated channel (subscription 122). The message will be uncompacted.

Table 18 Crossing RFQ Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
Header (8 bytes)					
Message Length	0	2	Binary		28
Message Type	2	1	Alpha/Numeric	AQ_MSG_TYPE	'j'
Subscription	3	1	Binary	AQ_SUBSCRIPTION	122
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since Midnight
Message Body (16 Bytes)					
Series Index	8	4	Binary	AQ_SERIES_INDEX	Symbol Mapping Index
Message Sequence Number	12	4	Binary	AQ_SEQUENCE	ARCA assigned sequence number. 1 – 4,294,967,294 within a series.
Price	16	4	Binary	AQ_PRICE	Price of the crossing transaction
Shares	20	4	Binary	AQ_ QUANTITY	Shares
cType	24	1	Char		Type of auction message, P = beginning of auction, G = In Auction, Q = End of Auction
Side	25	1	Char		Side of the RFQ 1 – Buy 2 – Sell
Filler	26	2	Char		Reserved for future use.

4.2.5 System Event Message

System Event messages are used to clear option quotes and send halt notifications.

Table 19 System Event Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
Header (8 bytes)					
Message Length	0	2	Binary		24
Message Type	2	1	Alpha/Numeric	AQ_MSG_TYPE	‘v’
Subscription	3	1	Binary	AQ_SUBSCRIPTION	0-255 identifying the Subscription
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since Midnight
Message Body (16 Bytes)					
Series Index	8	4	Binary	AQ_SERIES_INDEX	Series Mapping Index
Market ID	12	2	Binary	AQ_MARKET_ID	NULL
System ID	14	1	Binary	AQ_SYSTEM_ID	NULL
Bit	15	1	Alpha	AQ_BIT	NULL
Message Sequence Number	16	4	Binary	AQ_SEQUENCE	NYSE Arca assigned sequence number. 1 – 4,294,967,294 within a series. 0 - if the message clears quotes by Underlying or by Subscription
Reserved5	20	2	Alpha	AQ_RESERVE_5	NULL
Event Code	22	1	Alpha	AQ_EVENT_CODE	■ ‘A’ - Clear Series Ask ■ ‘B’ - Clear Series Bid ■ ‘C’ - Clear Series ■ ‘D’ - Clear Underlying Ask ** ■ ‘E’ - Clear Underlying Bid ** ■ ‘F’ - Clear Underlying ** ■ ‘G’ - Clear Subscription Ask ■ ‘H’ - Clear Subscription Bid ■ ‘I’ - Clear Subscription

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
					■ ‘L’ - Light up a dark series ■ ‘N’ - Open Indication (Dark series) ■ ‘O’ - Series is Open ■ ‘o’ - Open Indication (Underlying) ■ ‘S’ - Suspend (Halt Series) ■ ‘T’ - Unhalt Dark Series ■ ‘U’ - Unhalt Series. ■ ‘s’ – Suspend (Halt Underlying) ■ ‘u’ - Unhalt Underlying. ■ ‘X’ - Series is Closed ■ ‘x’ - Close Indication (Underlying) ** reserved for explicit symbology *Dark series is a series will not be quoted if there have not been any trades in that Series for specific number of days
Reset Code	23	1	Alpha	AQ_RESET_CODE	■ ‘C’ - Continue using existing sequence numbers. ■ ‘R‘- Reset Packet Sequence and reset all expected Message Sequence Numbers to 1 for this subscription. This message will be in a Packet Sequence Number of 1. This will be the first message of the day, per subscription. This message will also be used during a System Failover. This will only be used in conjunction with an Event

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	FAST TEMPLATE	DESCRIPTION
					Code of 'I'.

Note: In the event of a stock halt or unhalt condition, each individual Option Series will receive a halt or unhalt message. This message is independent of the stock halt or unhalt message for the underlying security. Both messages should be processed by NYSE ArcaBook users. See **Table 20** for Series Status message frequency.

Table 20 Crossing RFQ Message Format

CONDITION	FREQUENCY
Stock Opens	One message is sent, covers for all series
Stock Closes	One message is sent, covers for the stock
Stock Halts	One message is sent, covers for all series
Stock Unhalts	One message is sent, covers for all series
Series Opens	A message is sent for each **individual** series
Series Closes	One message is sent, covers for all series
Series Halts	One message is sent, covers for all series
Series UnHalts	A message is sent for each **individual** series

4.3 COMPLEX OPTIONS

4.3.1 Header Format

All NYSE ArcaBook for Complex Options messages are sent over multicast. Once uncompacted, they all begin with the following header.

Table 21 Header Format

HEADER	OFFSET	LEN	TYPE	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	0 – 65535 (value includes 8 byte header)
Message Type	2	1	Alpha/Numeric	A single character to identify the message: ■ 'c' - complex quote ■ 'd' - complex leg definition
Subscription	3	1	Binary	124 identifying the Subscription.
Time Stamp	4	4	Binary	Milliseconds since Midnight

4.3.2 Complex Option Leg Definition Message

The Leg Definition message indicates the parts of an option spread. Leg Definition message will reference a unique complex series ID. In this release an alphanumeric complex symbol is being added to the leg definition which will be valid until one of the legs expires.

A single Leg can be defined as an option series leg or a stock leg. This is indicated in the LegSecurityType field.

For Complex Options Leg Definitions, multiple messages can be included in a single packet. The number of Leg Definitions is provided in the message.

The entire complex leg definition will be encoded as AQ_BITMAP.

Table 22 Leg Definition Message Format

MESSAGE	OFFSET	LEN	TYPE	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	32+ - multiple legs definition can be defined
Message Type	2	1	Alpha/Numeric	‘d’
Subscription	3	1	Binary	124 identifying the Subscription
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body (40+ Bytes)				
Complex ID	8	8	Binary	NYSE Arca assigned complex ID.
NumLegs	16	1	Binary	Indicates number Legs of 1 - 255
Pad	17	7	Alpha	Null
Symbol	24	21	Alpha	Complex Symbol. This is an alpha numeric symbol which can be used for a unique strategy until one of the legs expire.
Pad	45	3	Alpha	Null
Repeating Leg Definition (16+ Bytes)				
Index	48	4	Binary	Stock or Series Index
Market ID	52	2	Binary	NULL
System ID	54	1	Binary	NULL
Bit	55	1	char	NULL
Leg Ratio Quantity	56	4	Binary	Leg Ratio
Leg Side	60	1	Alpha	‘B’ - Bid ‘S’ - Offer
Leg Security Type	61	1	Alpha	‘O’ - Option Series Leg ‘E’ - Equity Stock Leg
Pad	62	2	Alpha	Null

4.3.3 Complex RFQ Message

This message is sent out at the beginning, during, and end of auction of a complex order (for both Complex Order Auctions and Complex Crossing Auctions). The messages are sent through the Complex Consolidated channel (subscription 123).

Table 23 Complex RFQ Message Format

MESSAGE	OFFSET	LEN	TYPE	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	32
Message Type	2	1	Alpha/Numeric	‘k’
Subscription	3	1	Binary	128
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body (32+ Bytes)				
Complex ID	8	8	Binary	NYSE Arca-assigned Complex ID. See Complex ID
Message Sequence Number	16	4	Binary	ARCA assigned sequence number. 1 – 4,294,967,294 within a series.
Price	20	4	Binary	Price of the Crossing order Note: Price will not be displayed for COA
Shares	24	4	Binary	Shares

cType	28	1	Char	Type of RFQ Message:
				P = Begin Auction
				G = In Auction
				Q = End Auction

Side	29	1	Char	Side of the RFQ 1 – Buy 2 – Sell
Filler	30	2	Char	Reserved for future use

4.3.4 Top Quote Message

Top-of-Book identifies one price-level for one side of the NYSE ArcaBook for Complex Options. This is the same method used in NYSE ArcaBook for Options Aggregated Quote Message for both top and depth. Each Quote Message has two Price-Level fields identifying the price-level index and the operation that the message indicates for that level:

- Delete Price-Level = the index of the price-level that should be deleted
- Insert Price-Level = the index of the price-level that should be inserted

The Top of Book quote for complex will always modify price-level 1 (for the top of the book).

The Message Sequence Number is on a per–complex series id/subscription basis. Each complex series has a set of sequence numbers so that if a packet is lost, the book can still be updated without interruption for any complex series that is not missing messages because of lost packets. Quote Messages and System Event Messages will use the same Message Sequence Numbers because they are on the same subscription. See TCP/IP Recovery Messages for information on recovering missing packets.

Table 24 Top Quote Message Format

MESSAGE	OFFSET	LEN	TYPE	FAST TEMPLATE	DESCRIPTION
Header (8 bytes)					
Message Length	0	2	Binary		32
Message Type	2	1	Alpha/Numeric	AQ_MSG_TYPE	'c'
Subscription	3	1	Binary	AQ_SUBSCRIPTION	124 identifying the Subscription
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since Midnight
Message Body (32 Bytes)					
Complex ID	8	8	Binary	See Complex ID	NYSE Arca–assigned complex series id. 1 – 4,294,967,294 across all series.
Message Sequence Number	16	4	Binary	AQ_SEQUENCE	NYSE Arca assigned sequence number. 1 – 4,294,967,294 within a series.
Reserve6	20	4	Binary	AQ_RESERVE_6	NULL
Customer Contracts	24	4	Binary	AQ_CUSTOMER_VOLUME	Aggregated customer volume
Total Contracts/ Shares	28	4	Binary	AQ_QUANTITY	Aggregated volume of the quote
Price	32	4	Binary	AQ_ PRICE	The price for this quote level. 4 decimal places. This can be positive for credit or negative for debit.
Delete Price Level	36	1	Alpha	AQ_DELETE_LEVEL	Price-level index to delete for this quote. Always 1 for now.
Insert Price Level	37	1	Binary	AQ_INSERT_LEVEL	New price-level index for this quote. Always 1 for now.
Side	38	1	Alpha	AQ_BUY_SELL	'B' - Bid 'S' - Offer

MESSAGE	OFFSET	LEN	TYPE	FAST TEMPLATE	DESCRIPTION
Request For Quote Indicator	39	1	Binary	AB_RFQ	Not currently in use. This value will always be set to 0.

4.3.4.1 Top of Book Messaging

Table 25 Scenario 1: Top of Book Improves Market

In this scenario, the Top of Book message overwrites the previous quote and improves the market.

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	QUANTITY
Top Bid Quote	1	1	5	100
		PRICE LEVEL	**PRICE**	**QUANTITY**
Original Top of Book		1	6	200
Modified Top of Book		1	5	100

Table 26 Scenario 2: Top of Book message clears the quote

In this scenario, the Top of Book message overwrites the previous top of book with a zero quantity, zero price quote.

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	QUANTITY
Top Bid Quote	1	1	0	0
		PRICE LEVEL	**PRICE**	**QUANTITY**
Original Top of Book		1	6	200
Modified Top of Book		1	0	0

Table 27 Scenario 3: Customer improves Top of Book

In this scenario, the customer order overwrites the previous quote and improves the market.

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	QUANTITY	CUSTOMER VOLUME
Top Bid Customer Order	1	1	5		100
		PRICE LEVEL	**PRICE**	**QUANTITY**	**CUSTOMER VOLUME**
Original Top of Book		1	6	200	0

	DELETE PRICE LEVEL	INSERT PRICE LEVEL	INSERT PRICE	QUANTITY	CUSTOMER VOLUME
Modified Top of Book		1	5	100	100

4.3.5 System Event Message

System Event messages are used to clear option quotes and send halt notifications.

Table 28 System Event Message Format

MESSAGE	OFFSET	LEN	TYPE	FAST TEMPLATE	DESCRIPTION
Header (8 bytes)					
Message Length	0	2	Binary		24
Message Type	2	1	Alpha/Numeric	AQ_MSG_TYPE	‘e’
Subscription	3	1	Binary	AQ_SUBSCRIPTION	124 identifying the Subscription
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since Midnight
Message Body (16 Bytes)					
Complex ID	8	8	Binary	See Complex ID	NYSE Arca–assigned complex series ID.
Message Sequence Number	16	4	Binary	AQ_SEQUENCE	NYSE Arca-assigned sequence number. 1 – 4,294,967,294 within a series.
Reserve5	20	2	Alpha	AQ_RESERVE_5	NULL
Event Code	22	1	Alpha	AQ_EVENT_CODE	‘A’ - Clear Ask Series ‘B’ - Clear Bid Series ‘C’ - Clear Series ‘D’ - Clear Ask Underlying ‘E’ - Clear Bid Underlying ‘F’ - Clear Underlying ‘G’ - Clear Subscription Ask ‘H’ - Clear Subscription Bid ‘I’ - Clear Subscription ‘S’ – Halt (Suspend) Series ‘U’ - UnHalt Series ‘s’ - Halt (Suspend) Underlying ‘u’ - UnHalt Underlying

MESSAGE	OFFSET	LEN	TYPE	FAST TEMPLATE	DESCRIPTION
					‘O’ - Open Series ‘X’ - Close Series ‘o’ - Open Underlying ‘x’ - Close Underlying
Reset Code	23	1	Alpha	AQ_RESET_CODE	‘C’ - Continue using existing sequence numbers. ‘R’ - Reset *Packet Sequence* and reset all expected *Message Sequence Numbers* to 1 for this subscription. This message will be in a Packet Sequence Number of 1. This will be the first message of the day, per subscription. This message will also be used during a System Failover. This will only be used in conjunction with an Event Code of ‘I’.

4.4 TRADES

4.4.1 Header Format

All Trades messages are sent over multicast. They all begin with the following header.

Table 29 Header Format

MESSAGE HEADER	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	0 – 65535 (value includes 8 byte header)
Message Type	2	1	Alpha/Numeric	A single character to identify the message
Possible Duplicate	3	1	Binary	'1' - possible duplicate message
Time Stamp	4	4	Binary	Milliseconds since Midnight

4.4.2 Last Sale Message

This message is sent for either of these trade events:

- An order partially trades
- An order completely trades

Table 30 Last Sale Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	40
Message Type	2	1	Alpha/Numeric	'x'
Subscription	3	1	Binary	127 identifying the Subscription
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body (32 bytes)				
Series Index	8	4	Binary	Series Mapping Index.
Market ID	12	2	Binary	NULL
System ID	14	1	Binary	NULL
Bit	15	1	Alpha	NULL
Sequence Number	16	4	Binary	Message sequence number. 1 – 4,294,967,294
Contracts	20	4	Binary	Number of contracts traded
Trade Reference Number	24	8	Binary	The unique reference number per trading platform (system code) assigned to this trade.

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Price	32	4	Binary	Trade price
Possible Duplicate	36	1	Binary	‘0’ - original message ‘1’ - possible duplicate message
Complex	37	1	Alpha	‘P’ - Complex trade with equity leg ‘L’ - Complex trade
Sale Condition	38	1	Alpha	‘I’ - Late Report ‘R’ - Floor Trade ‘S’ - ISO Sweep Trade
Padding	39	1	Alpha	Padding

4.4.3 Trade Bust or Correction Message

This message is sent for trade busts or trade corrections. The Event Code field identifies the triggering event.

Table 31 Trade Bust or Correction Message Format

MESSAGE	OFFSET	SIZE (BYTES)	TYPE	NOTES AND VALUES
Header (8 bytes)				
Message Length	0	2	Binary	40
Message Type	2	1	Alpha/Numeric	‘u’
Subscription	3	1	Binary	127 identifying the Subscription
Time Stamp	4	4	Binary	Milliseconds since Midnight.
Message Body (32 bytes)				
Series Index	8	4	Binary	Series Mapping Index.
Market ID	12	2	Binary	NULL
System ID	14	1	Binary	NULL
Bit	15	1	Alpha	NULL
Sequence Number	16	4	Binary	Message sequence number. 1 – 4,294,967,294
Contracts	20	4	Binary	Number of contracts traded
Trade Reference Number	24	8	Binary	The unique reference number per trading platform (system code) assigned to the trade that has been busted or corrected.
Price	32	4	Binary	Price of the trade that has been busted or corrected. For corrections, this represents the

MESSAGE	OFFSET	SIZE (BYTES)	TYPE	NOTES AND VALUES
				corrected price.
Possible Duplicate	36	1	Binary	0 - original message 1 - possible duplicate message
Complex	37	1	Alpha	‘P’ - Complex trade with equity leg ‘L’ - Complex trade
Sale Condition	38	1	Alpha	‘S’ - ISO Sweep Trade ‘I’ - Late Report
Event Code	39	1	Alpha	‘B’ - trade bust ‘C’ - trade correction

4.4.4 System Event Message

System Event messages are used to Halt and Unhalt a Series.

Table 32 System Event Message Format

MESSAGE	OFFSET	SIZE (BYTES)	TYPE	NOTES AND VALUES
Header (8 bytes)				
Message Length	0	2	Binary	24
Message Type	2	1	Alpha/Numeric	‘w’
Subscription	3	1	Binary	0-255 identifying the Subscription
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body (16 Bytes)				
Series Index	8	4	Binary	Series Mapping Index
Market ID	12	2	Binary	NULL
System ID	14	1	Binary	NULL
Bit	15	1	Alpha	NULL
Message Sequence Number	16	4	Binary	NYSE Arca assigned sequence number. 1 – 4,294,967,294 within a series.
Reserved	20	2	Alpha	NULL
Event Code	22	1	Alpha	S - Suspend(Halt Series) U - Unhalt Series.
Reserved	23	1	Alpha	NULL

5. TCP/IP RECOVERY MESSAGES

5.1 NYSE ARCABOOK FOR OPTIONS RECOVERY SERVER MESSAGES

5.1.1 Heartbeat Request Message

The Recovery Server sends this message every 60 seconds. This prevents some firewalls from timing out the TCP/IP connection. Clients must respond with a Heartbeat Response message. This message only has a message header.

Table 33 Heartbeat Request Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	8 bytes
Message Type	2	1	Alpha/Numeric	‘h’
Padding	3	1		
Time Stamp	4	4	Binary	Milliseconds since Midnight

5.1.2 Test Response Message

The Recovery Server sends this message in response to a Test Request message from a client.

Table 34 Test Response Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	28 bytes
Message Type	2	1	Alpha/Numeric	‘t’
Padding	3	1		
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body (20 Bytes)				
Test Message	8	20	Alpha	The client text to echo from the Test Request message

5.1.3 Login Accepted Message

The Recovery Server sends this message to indicate that a client's login request has been accepted. For NYSE ArcaBook for Options, the message includes how many price levels are currently included in the Depth of Book data feed.

Table 35 Login Accepted Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	12 bytes
Message Type	2	1	Alpha/Numeric	‘l’
Padding	3	1		
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body (4 Bytes)				
Padding	8	4	Alpha	

5.1.4 Login Rejected Message

The Recovery Server sends this message when a client request to log in is invalid. This message is also sent when the server has exceeded the maximum connection limit for this port or when a connection has timed out (client connects and does not log in within 30 seconds). The Reject Code field indicates the reason for the rejection. The Recovery Server closes the socket connection after sending this message.

Table 36 Login Rejected Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	12 bytes
Message Type	2	1	Alpha/Numeric	‘r’
Padding	3	1		
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body (4 Bytes)				
Reject Code	8	1	Alpha	■ ‘A’ - Not Authorized ■ ‘M’ - Maximum Server Connections Reached ■ ‘T’- Timeout
Padding	9	3	Alpha	

5.1.5 Packet Replay Message

The Recovery Server sends this message in response to client requests for missing packets.

Table 37 Packet Replay Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Packet Length	0	2	Binary	Size of Compacted Messages
Message Type	2	1	Alpha	See Packets for more information. ■ M - Message ■ N - Not found if the packet requested is unknown
Subscription	3	1	Binary	0-255 identifying the Subscription
Packet Sequence Number	4	4	Binary	Sequence Number of Packet
Message Body				
Compacted Messages[1]	8	Variable	FIX FAST Compacted	Each quote message within a packet also has a message sequence number. Message sequence numbers start from one each day and are incremented by one per unique complex option ID. Message sequence numbers can be used to facilitate the continued processing of the stream while the dropped packets are being recovered. See UDP Multicast Messages for more information.

[1] This is uncompacted for Trades

5.1.6 Complex Options: Leg Definition Replay Message

The NYSE ArcaBook for Options Recovery Server sends this message in response to client requests for leg definitions. This is same leg definition message that was sent in the UDP feed with the exception that the message is sent back via TCP/IP and is not compressed. See Complex Option Leg Definition Message.

Table 38 Leg Definition Replay Message Format

MESSAGE	OFFSET	LEN	TYPE	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	32+ - multiple legs definition can be defined
Message Type	2	1	Alpha/Numeric	‘d’
Subscription	3	1	Binary	124 identifying the Subscription

MESSAGE	OFFSET	LEN	TYPE	DESCRIPTION
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body (40+ Bytes)				
Complex ID	8	8	Binary	NYSE Arca assigned complex ID.
NumLegs	16	1	Binary	Indicates number Legs of 1 - 255
Pad	17	7	Alpha	Null
Symbol	24	21	Alpha	Complex Symbol. This is an alpha numeric symbol which can be used for a unique strategy until one of the legs expire.
Pad	45	3	Alpha	Null
Repeating Leg Definition (16+ Bytes)				
Index	48	4	Binary	Stock or Series Index
Market ID	52	2	Binary	NULL
System ID	54	1	Binary	NULL
Bit	55	1	char	NULL
Leg Ratio Quantity	56	4	Binary	Leg Ratio
Leg Side	60	1	Alpha	‘B’ - Bid ‘S’ - Offer
Leg Security Type	61	1	Alpha	‘O’ - Option Series Leg ‘E’ - Equity Stock Leg
Pad	62	2	Alpha	Null

5.2 MESSAGES SENT BY A SUBSCRIBER

5.2.1 Login Request Message

Clients send this message to log into the Recovery Server. The server responds with a Login Accepted or Login Rejected message.

Table 39 Login Request Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8bytes)				
Message Length	0	2	Binary	28 bytes
Message Type	2	1	Alpha/Numeric	‘L’

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Padding	3	1		
Time Stamp	4	4	Binary	Milliseconds since midnight (not required)
Message Body (20 Bytes)				
Username	8	8	Alpha	Username
Password	16	12	Alpha	Password

5.2.2 Heartbeat Response Message

Clients send this message as a response to the Heartbeat Request message. If the server does not receive a Heartbeat Response within 60 seconds of sending the Heartbeat Request message, the server closes the connection. This message only contains a header.

Table 40 Heartbeat Response Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Message Length	0	2	Binary	8 bytes
Message Type	2	1	Alpha/Numeric	‘H’
Padding	3	1		
Time Stamp	4	4	Binary	Milliseconds since midnight (not required)

5.2.3 Index Mapping Request Message

Clients send this message to request a response from the Recovery Server to get a full mapping of underlying stock and series details. To get a full mapping with a single request, you can use the series-status subscription number (122). The mapping messages are sent over the TCP socket uncompacted for all subscriptions.

Table 41 Index Mapping Request Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	8 bytes
Message Type	2	1	Alpha/Numeric	‘M’
Subscription	3	1	Binary	0 -255, the multicast subscription number
Time Stamp	4	4	Binary	Milliseconds since midnight (not required)

5.2.4 Test Request Message

Clients send this message to request a response from the Recovery Server during periods of inactivity. The client can specify a text message for the server to echo backing its response.

Table 42 Test Request Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	28 bytes
Message Type	2	1	Alpha/Numeric	‘T’
Padding	3	1		

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Time Stamp	4	4	Binary	Milliseconds since midnight (not required)
Message Body (20 Bytes)				
Test Message	8	20	Alpha	Text to be echoed

5.2.5 Complex Options: Leg Definitions Request Message

Clients request leg definitions with this message. Client can request all definitions for the specific subscription by specifying a 0 (zero) in the Complex ID field or clients can request a single definition by specifying a valid Complex ID (1 – N).

Table 43 Leg Definitions Message Format

MESSAGE	OFFSET	LEN	TYPE	DESCRIPTION
Header (8 bytes)				
Message Length	0		Binary	16 bytes
Message Type	2	1	Alpha/Numeric	‘D’
Subscription	3	1	Binary	124, the multicast subscription number
Time Stamp	4	4	Binary	Milliseconds since Midnight (not required)
Message Body (8 Bytes)				
Complex ID	8	8	Binary	Complex Series ID.

5.2.6 Dropped Packet Request Message

Clients request missing packets with this message. This can be a single packet or a contiguous set of packets. Packets are identified by the Subscription number (multicast address) and the packet number.

Table 44 Dropped Packet Request Message Format

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Header (8 bytes)				
Message Length	0	2	Binary	16 bytes
Message Type	2	1	Alpha/Numeric	‘P’
Subscription	3	1	Binary	0 -255, the multicast subscription number for this missed packet
Time Stamp	4	4	Binary	Milliseconds since midnight (not required)
Message Body (8 Bytes)				
Starting Packet Number	8	4	Binary	1 – 4,294,967,294

MESSAGE	OFFSET	SIZE (BYTES)	FORMAT	DESCRIPTION
Ending Packet Number	12	4	Binary	1 – 4,294,967,294

6. FIX FAST PROTOCOL

This chapter applies to options only, not trades.

6.1 OVERVIEW

Subscribers receive the NYSE ArcaBook for Options real-time data feed in the FAST Protocol. This protocol is a standard method for compacting real-time market data resulting in reduced bandwidth. The complete FAST specification is available at:
http://fixprotocol.org/documents/1766/FAST%20SERDES%20Specification%200.5%202005-07-28.zip and
http://fixprotocol.org/documents/1536/BMF%20Specification%200.14.zip

The FAST Protocol uses two main approaches to reduce bandwidth:

- **Omit Redundant Fields** This approach uses two FAST features:
  - FAST Templates that specify the FAST field encoding to control field omission and reconstitution. Field encoding schemes define whether fields can be omitted and how they should be interpreted if omitted.

    For example, Copy encoding specifies that if a field is not present, you should use a copy of the field from the previous message. Increment encoding specifies that you should use the previous value and increment it by some constant (usually 1). A field defined with an encoding scheme of None means that it will always be present.
  - Presence Map that indicates which fields are actually present in a message.
- **Variable Length Fields** That compact the bits used to represent a field's value. This uses continuation bit encoding to separate the fields. Only the first seven bits of a byte transmit data. The high bit is the continuation bit that indicates whether data for the field continues or stops. When the high bit is set, this is called a stop bit and indicates the end of the variable length field.

6.1.1 A FAST Message

A FAST message consists of a minimum of a one-byte Presence Map (pmap) followed by zero or more bytes of field data, as shown below:

FastMessage := ::= < pmap { pmap} > < { field } >

The pmap may be more than one byte and also uses continuation bit encoding (it ends in a stop bit). The pmap sets individual bits to either 1 or 0 to indicate if a specific field is present in the FAST message.

A field within a FAST message can represent one of four data types:

- Signed integer
- Unsigned integer
- ASCII string
- Bitmap

All fields are variable length, ending in a stop bit.

6.2 THE NYSE ARCABOOK FOR OPTIONS FAST IMPLEMENTATION

The NYSE ArcaBook for Options FAST implementation reduces bandwidth requirements by up to 80%, or a ratio of 5 to 1. Each message within the FAST NYSE ArcaBook for Options data feed has a minimum of two bytes: a Presence Map of at least one byte and a Message Type field of one byte. Note that there may be more than one byte in the pmap, but there will always be at least one. The encoding scheme of None for the message type field guarantees that it will be present in every message.

6.3 SAMPLE SOURCE CODE

To help subscribers process the NYSE ArcaBook for Options FAST feed, NYSE Arca provides source code to decode NYSE ArcaBook for Options FAST messages into NYSE ArcaBook for Options binary messages. A single, C language routine, AQFastDecodeComplex() , decodes NYSE ArcaBook for Options FAST messages. The following pseudo code describes the decoding process.

```
Define some variables to hold our input buffer and results
Integer length
Integer result
Byte buffer[2048]
ArcaBookOptionsMessage message;

Process until we are told to stop
Do

    Call the decode routine, we decode the FAST message in
    “buffer” and place the result in “message”, “length” will
    contain the number of bytes we processed in “buffer”.
    result = AQFastDecode(buffer, length, message)

    Check the result code
    If result == AB_OK Then
         process the ArcaBookOptionsBinary message, and
         advance the buffer to buffer + length
         ProcessMessage(message)
    Else If result == AQ_INCOMPLETE_ERROR Then
         buffer did not contain a full FAST message, so
         read more bytes from the Multicast or TCP Recovery socket
         and place the result into buffer
         length = SocketRead(buffer,1024)
    Else
         We encountered some other error
         ProcessError(result)
    End

While Stop == False
```

This pseudo code is a very basic example. Please see the provided C source code for a full, working example.

6.4 TEMPLATE INFORMATION

The FAST template for each message indicates which fields may be omitted from a message and how clients should interpret omitted fields. NYSE ArcaBook FAST messages use the message type as the FAST template ID. Once clients have parsed the message type, the rest of the message can be parsed based on the template shown in **Table 45**.

6.4.1 ArcaBook for Options

Table 45 NYSE ArcaBook for Options FAST Message Template

FIELD ID	FIELD NAME	IN MESSAGES OF TYPE	FAST TYPE	ENCODING
0	AQ_MSG_TYPE	All	Unsigned8	None
1	AQ_SUBSCRIPTION	All	Unsigned8	Copy
2	AQ_TIME	All	Unsigned32	Copy
3	AQ_SEQUENCE	q, i, v	Unsigned32	Increment
4	AQ_SERIES_INDEX	q, i, v, m,	Unsigned32	Copy
5	AQ_UNDERLYING_INDEX	m, n,	Unsigned32	Copy
	AQ_MKT_IMBALANCE	i,		
6	AQ_ PRICE	q, i,	Unsigned32	Copy
7	AQ_UNDERLYING_QUANTITY	m,	Unsigned32	Copy
	AQ_QUANTITY	q, i,		
8	AQ_CUSTOMER_VOLUME	q,	Unsigned32	Copy
	AQ_TOTAL_IMBALANCE	i,		
9	AQ_MARKET_ID	All	Unsigned16	Copy
10	AQ_INSERT_LEVEL	q,	Unsigned8	Copy
11	AQ_DELETE_LEVEL	q,	Unsigned8	Copy
12	AQ_SYSTEM_ID	All	Unsigned8	Copy
13	AQ_PRICE_SCALE	q, m, n,	Unsigned8	Copy
14	AQ_SYMBOL	m, n,	ascii	Copy
15	AQ_EXPIRE_YEAR	m,	ascii	Copy
16	AQ_EXPIRE_MONTH	m,	ascii	Copy
17	AQ_EXPIRE_DAY	m,	ascii	Copy
18	AQ_STRIKE_PRICE	m,	ascii	Copy
19	AQ_STRIKE_DECIMAL	m,	ascii	Copy
20	AQ_OCC_SYMBOL	m,	ascii	Copy
21	AQ_PUT_CALL	m,	char	Copy
	AQ_SECURITY_TYPE	n,		
	AQ_BUY_SELL	q,		
	AQ_EVENT_CODE	v,		

FIELD ID	FIELD NAME	IN MESSAGES OF TYPE	FAST TYPE	ENCODING
	AQ_AUCTION_TYPE	i,		
22	AQ_BIT	All	char	Copy
23	AQ_PRICE_RESOLUTION	n,	char	Copy
	AQ_RESET_CODE	v,		
24	AQ_EXCHANGE_CODE	n,	char	Copy
25	AQ_RESERVE_1	m,	Unsigned16	Copy
	AQ_AUCTION_TIME	i,		
26	AQ_RESERVE_2	m,	Unsigned8	Copy
27	AQ_RESERVE_3	m, n,	char	Copy
28	AQ_RESERVE_4	i, m, n,	char	Copy
29	AQ_RESERVE_5	v,	ascii	Copy
30	AQ_RESERVE_6	q, m,	Unsigned32	Copy
31	AQ_BITMAP	none	Bitmap	None

Note: Field IDs with multiple Field Names are guaranteed never to occur more than once in a given message.

6.4.2 ArcaBook for Complex Options

Table 46 NYSE ArcaBook for Complex Options FAST Message Template

FIELD ID	FIELD NAME	IN MESSAGES OF TYPE	FAST TYPE	ENCODING
0	AB_MSG_TYPE	m, n, c, e,	Unsigned8	None
1	AB_SUBSCRIPTION	m, n, c, e,	Unsigned8	Copy
2	AB_TIME	m, n, c, e,	Unsigned32	Copy
3	AB_SEQUENCE	c, e,	Unsigned32	Increment
4	AB_SERIES_INDEX	m,	Unsigned32	Copy
	AB_ORDER_ID	c, e,		
5	AB_UNDERLYING_INDEX	m, n,	Unsigned32	Copy
	AB_COMPLEX_ID	c, e,		
6	AB_ PRICE	c,	Unsigned32	Copy
7	AB_UNDERLYING_QUANTITY	m,	Unsigned32	Copy
	AB_QUANTITY	c,		
9	AB_MARKET_ID	m, n, c, e,	Unsigned16	Copy
10	AB_INSERT_LEVEL	c,	Unsigned8	Copy

FIELD ID	FIELD NAME	IN MESSAGES OF TYPE	FAST TYPE	ENCODING
11	AB_DELETE_LEVEL	c,	Unsigned8	Copy
12	AB_SYSTEM_ID	m, n, c, e,	Unsigned8	Copy
13	AB_PRICE_SCALE	m, n,	Unsigned8	Copy
	AB_RFQ	c,		
14	AB_SYMBOL	m, n,	ascii	Copy
15	AB_EXPIRE_YEAR	m,	ascii	Copy
16	AB_EXPIRE_MONTH	m,	ascii	Copy
17	AB_EXPIRE_DAY	m,	ascii	Copy
18	AB_STRIKE_PRICE	m,	ascii	Copy
19	AB_STRIKE_DECIMAL	m,	ascii	Copy
20	AB_OCC_SYMBOL	m,	ascii	Copy
21	AB_PUT_CALL	m,	char	Copy
	AB_SECURITY_TYPE	n,		
	AB_BUY_SELL	c,		
	AB_EVENT_CODE	e,		
22	AB_BIT	m, n, c, e,	char	Copy
23	AB_PRICE_RESOLUTION	n,	char	Copy
	AB_RESET_CODE	e,		
24	AB_EXCHANGE_CODE	n,	char	Copy
25	AB_RESERVE_1	n,	Unsigned16	Copy
26	AB_RESERVE_2	n,	Unsigned8	Copy
27	AB_RESERVE_3	n, m,	char	Copy
28	AB_RESERVE_4	n, m,	char	Copy
29	AB_RESERVE_5	e,	ascii	Copy
30	AB_RESERVE_6	c,	Unsigned32	Copy
31	AB_BITMAP	d,	Bitmap	None

Note: Field IDs with multiple Field Names are guaranteed never to occur more than once in a given message.

APPENDIX A: CHANGE HISTORY – PREVIOUS VERSIONS

A.1 NYSE ARCABOOK FOR OPTIONS CLIENT SPECIFICATION VERSION HISTORY

The following table describes the changes that were made to the standalone *NYSE ArcaBook for Options Client Specification* before it was amalgamated into the current Client Specification. The last version of the standalone document was v.3.14 as described in the table.

VERSION NO.	DATE	CHANGE DESCRIPTION
1.0	01/30/2006	Updates to length of OCC Option Symbol field in all messages. First published version.
1.1	02/13/2006	Spec name change. Copy edits to make Intro section more like other specs. Draft FAST template
	03/01/2006	Modify FAST template section.
1.2	03/28/2006	A Clearbook Reset has a next expected Packet Sequence Number of 2. Added Message Sequence Number to System Event Message. Indicate possible internal usage of underlying symbol field.
1.3	04/04/2006	Insert of Partial Book Example(Scenario 5)
1.4	05/03/2006	System Event (Halt/Unhalt) Tentative Multicast IP Breakout
1.5	10/18/2006	Add System Event codes for open, close, halts, and unhalts. Add IPs and ports for new series status (subscription 253).
1.6	11/01/2006	Added Note concerning halts/unhalts in System Events
1.7	01/17/2007	Penny Pilot Subscription Rebalance
1.8	01/26/2007	Additional Event codes added.
1.9	09/07/2007	New allocations of issues by subscription (Chapter 2 – Communications)
2.0	09/12/2007	Updated allocations of issues by subscription (Chapter 2 – Communications)
2.1	10/16/2007	Updated allocations of AAPL by subscription (Chapter 2 – Communications)
2.2	10/29/2007	Updated allocations of IWM and DIA by subscription (Chapter 2 – Communications)
2.3	01/25/2008	Updated allocations of AAPL, GOOG and Penny Pilot by subscription (AAPL moved to subscriptions 3 & 7; GOOG moved to 24 & 28; Penny Pilot moved out of 24 & 28) (Chap 2 – Communications)
3.0	10/17/2008	New Underlying and Series Index Message Index Mapping Request. New Message Formats, New IPs.
3.01	11/14/2008	Included Old IPs
3.02	12/12/2008	Update Fast Template

VERSION NO.	DATE	CHANGE DESCRIPTION
3.03	01/14/2009	Added source IP
3.04	01/22/2009	Added additional details to index mapping request. Made underlying quantity reserved for future use.
3.05	01/31/2009	IP address changes for subscription 124-127
3.06	02/06/2009	Modified Cert subscriptions
3.07	03/31/2009	Corrected IP for imbalance
3.08	10/23/2009	Symbology update
3.09	01/22/2010	Corrected Cert IPs
3.10	01/28/2010	Updates to document format and TCP Recovery Section
3.11	03/31/2010	Crossing RFQ update
3.12	12/13/2010	Fixed CERT IPs and updated formatting
3.13	02/21/2012	Updated Section 2.3 (AB Consolidated Channel)
3.14	04/25/2012	Minor updates to Appendix and throughout

A.2 NYSE ARCABOOK FOR COMPLEX OPTIONS CLIENT SPECIFICATION VERSION HISTORY

The following table describes the changes that were made to the standalone *NYSE ArcaBook for Complex Options Client Specification* before it was amalgamated into the current Client Specification. The last version of the standalone document was v.3.11 as described in the table.

VERSION NO.	DATE	CHANGE DESCRIPTION
1.0	04/09/2008	Added RFQ (Request For Quote) field to quote message. Changed encoding for Leg Definitions to BitMap. Indicate RFQ for future use.
1.1	06/06/2008	Added bandwidth information to Communications chapter.
1.2	07/23/2008	Indicate price can be positive or negative in Complex Top Quote Message.
1.3	08/27/2008	Add RFQ
3.0	10/14/2008	Update with Index Mapping and 64 bit Complex ID
3.01	11/14/2008	Included Old IPs
3.02	12/12/2008	Update fast Temple
3.03	01/14/2009	Added source IP
3.04	01/22/2009	Added additional details to index mapping request. Made underlying quantity reserved for future use.

VERSION NO.	DATE	CHANGE DESCRIPTION
3.05	01/31/2009	Changed the IPs, ports to reflect SFTI
3.06	09/14/2009	Modifications for Amex
3.06a	04/30/2010	Formatted into new template IP addresses removed and link to IP Addresses page added Bandwidth removed and link to Capacity page added
3.09	12/15/2010	Updated formatting
3.10	02/11/2011	Updated Access information in Chapter 2, “Communication”. Other changes made to: 3.1.3 Complex ID, 3.2 Packets, 3.3 Compaction, 4.5 Complex RFQ Message, and 4.6 Top of Quote Message.
3.11	04/24/2012	Minor updates to Appendix and throughout

A.3 NYSE ARCATRADE FOR OPTIONS CLIENT SPECIFICATION VERSION HISTORY

The following table describes the changes that were made to the standalone *NYSE ArcaTrade for Options Client Specification* before it was amalgamated into the current Client Specification. The last version of the standalone document was v.3.07 as described in the table.

VERSION NO.	DATE	CHANGE DESCRIPTION
1.00	04/03/2006	Updated bandwidth recommendation
1.05	05/03/2006	System Event(Halt/Unhalt) Tentative Multicast IP included.
1.06	04/14/2008	Cosmetic edits
3.00	10/17/2008	Added Sale Condition, Series Index Mapping, Removed OPRA symbology
3.01	01/14/2009	Included Old IPs Added source IP
3.02	01/21/2009	Added Cert IP. Change Underlying message type to ‘n’
3.03	01/22/2009	Added additional details to index mapping request. Made underlying quantity reserved for future use.
3.04	01/31/2009	Changed IPs for complex, Series, Imbalance, Trades channels
3.05	05/22/2009	Changed system event msg type from v to w
3.06	01/05/2010	Symbology Update
3.06a	04/29/2010	Formatted into new template IP addresses removed and link to IP Addresses page added

VERSION NO.	DATE	CHANGE DESCRIPTION
3.07	06/01/2011	Added value of ‘R’ to the Sale Condition field in Table 8 (Last Sale Message Format)