ArcaBook® Specification for Options

For NYSE Arca Options Exchange

June 4, 2008
Version 2.5

Revision History

Date	Revision	Change Made by:	Synopsis of Change
10-26-2005	Alpha 0.1	Chris McCown	Created Spec
11-04-2005	Alpha 0.2	Fred Jones	Additional Verbiage
11-09-2005	Alpha 0.3	Chris McCown Sara Mitchell	Removed Series Index. Edited for consistency
11-18-2005	Alpha 0.4	Chris McCown	Changed Packet Trailer and Compacted Delimiters and modified Message Header.
12-04-2005	Alpha 0.5	Chris McCown Sara Mitchell	Added more Clear Book Event codes. Edits for consistency and clarify
12-23-2005	Alpha 0.6	Chris McCown	Reformatted Header and updated Symbology
01-05-2006	Alpha 0.7	Chris McCown	Removed Trailer, Length includes header, Delete on a Partial Book Example
01-06-2006	Beta 0.8	Sara Mitchell	Copy edits. Added description for Corporate Action field and changed brand name (ArcaBook for Options).
01-17-2006	Beta 0.9	Sara Mitchell	Add OCC Option Symbol field and reorganized messages. Some copy edits. Add Not found packet message type.
01-30-2006	1.0	Sara Mitchell	Updates to length of OCC Option Symbol field in all messages. First published version.
02-13-2006	1.1	Sara Mitchell	Spec name change. Copy edits to make Intro section more like other specs. Draft FAST template
03-01-2006	1.1	Fred Jones	Modify FAST template section.
03-28-2006	1.2	Chris McCown	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.
04-04-2006	1.3	Chris McCown	Insert of Partial Book Example(Scenario 5)
05-03-2006	1.4	Chris McCown	System Event(Halt/Unhalt) Tentative Multicast IP Breakout
10-18-2006	1.5	Fred Jones	Add System Event codes for open, close, halts, and unhalts. Add ips and ports for new series status (subscription 253).
11-1-06	1.6	Ariel Rastatter	Added Note concerning halts/unhalts in System Events
01-17-07	1.7	Chris McCown	Penny Pilot Subscription Rebalance

1-26-07	1.8	Ariel Rastatter	Additional Event codes added.
9-7-07	1.9	Chris McCown	New allocations of issues by subscription (Chapter 2 – Communications)
9-12-07	2.0	Chris McCown	Updated allocations of issues by subscription (Chapter 2 – Communications)
10-16-07	2.1	Dzin Dzilna	Updated allocations of AAPL by subscription (Chapter 2 – Communications)
10-29-07	2.2	Dzin Dzilna	Updated allocations of IWM and DIA by subscription (Chapter 2 – Communications)
1-25-08	2.3	Dzin Dzilna	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) (Chapt 2 – Communications)
5-1-08 (changes effective 5-5-08)	2.4	Dzin Dzilna	Updated allocations of issues by subscription (Chapter 2 – Communications)
6-4-08 (changes effective 6-9-08)	2.5	Dzin Dzilna	Updated allocations of issues by subscription; changes in bold (Chapter 2 – Communications)

Table of Contents

1 Introduction

ArcaBook is a real-time data feed that disseminates equity, bond and option order/consolidated book information to subscribers for Archipelago exchanges. ArcaBook allows subscribers to produce and display the Archipelago® open order or consolidated book and ticker. Order routing algorithms can also use ArcaBook data.

Archipelago provides two ArcaBook interfaces to meet different customer requirements:

Interface	Equities	Bonds	Options	Description
ArcaBook for Equities and Bonds (formerly ArcaBook Binary)	✓	✓		A binary data feed or FIX Adapted for Streaming (FAST) compacted data feed for Listed, OTC, or Bulletin Board equity orders and/or bond orders.
ArcaBook for Options			✓	A FAST compacted data feed for aggregate option quotes of the consolidated book.

This specification is for developers that wish to write applications that interface with ArcaBook for Options.

ArcaBook Interface for Options

This API is message-based, using fixed length messages over the UDP IP protocol with binary numeric and fixed length ASCII fields. Binary values are in network Endian (Bid Endian) format.

Data Feeds

Customers can subscribe to two types of Aggregate Quotation 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.

Depth of Book Messaging

Archipelago 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 the Depth of Book Messaging section for examples.

Archipelago API Certification

Subscribers must certify their ArcaBook subscription clients with Archipelago. Archipelago provides an IP address, port number, username, and password to use for testing. To schedule a test, please call the FIX hotline at 888-689-7739 or email fix@archipelago.com.

2 Communication

Access

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

Multicast Quote Feeds

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

Each subscription address has two data feeds each with an IP address and port. For example:

Subscription	Type	Underlyings	Primary Multicast IP	Primary Port	Secondary Multicast IP	Secondary Port
0	Depth	AA* - AZPN	224.1.2.1	11000	224.1.2.33	12000
1	Depth	B - BZH	224.1.2.1	11001	224.1.2.33	12001
2	Depth	C - CZN	224.1.2.1	11002	224.1.2.33	12002
3	Depth	A* – CSCO and IWM Penny Pilots	224.1.2.1	11003	224.1.2.33	12003
4	Top	AA* - AZPN	224.1.2.2	11004	224.1.2.34	12004
5	Top	B - BZH	224.1.2.2	11005	224.1.2.34	12005
6	Top	C - CZN	224.1.2.2	11006	224.1.2.34	12006
7	Top	A* – CSCO and IWM Penny Pilots	224.1.2.2	11007	224.1.2.34	12007
8	Depth	D – FXP	224.1.2.3	11008	224.1.2.35	12008
9	Depth	GA – IYT	224.1.2.3	11009	224.1.2.35	12009
10	Depth	**JADE -MYL**	224.1.2.3	11010	224.1.2.35	12010
11	Depth	**DELL – MSFT** Penny Pilots	224.1.2.3	11011	224.1.2.35	12011
12	Top	D – FXP	224.1.2.4	11012	224.1.2.36	12012
13	Top	GA – IYT	224.1.2.4	11013	224.1.2.36	12013
14	Top	**JADE - MYL**	224.1.2.4	11014	224.1.2.36	12014
15	Top	**DELL – MSFT** Penny Pilots	224.1.2.4	11015	224.1.2.36	12015
16	Depth	**N – RZ**	224.1.2.5	11016	224.1.2.37	12016

17	Depth	S – TZOO	224.1.2.5	11017	224.1.2.37	12017
18	Depth	U – ZZ	224.1.2.5	11018	224.1.2.37	12018
19	Depth	**NEM -YHOO** Penny Pilot	224.1.2.5	11019	224.1.2.37	12019
20	Top	**N – RZ**	224.1.2.6	11020	224.1.2.38	12020
21	Top	S – TZOO	224.1.2.6	11021	224.1.2.38	12021
22	Top	U – ZZ	224.1.2.6	11022	224.1.2.38	12022
23	Top	**NEM -YHOO** Penny Pilot	224.1.2.6	11023	224.1.2.38	12023
24	Depth	GOOG**	224.1.2.7	11024	224.1.2.39	12024
25	Depth	DIA Penny Pilot	224.1.2.7	11025	224.1.2.39	12025
26	Depth	QQQQ Penny Pilot	224.1.2.7	11026	224.1.2.39	12026
27	Depth	SPY Penny Pilot	224.1.2.7	11027	224.1.2.39	12027
28	Top	GOOG**	224.1.2.8	11028	224.1.2.40	12028
29	Top	DIA Penny Pilot	224.1.2.8	11029	224.1.2.40	12029
30	Top	QQQQ Penny Pilot	224.1.2.8	11030	224.1.2.40	12030
31	Top	SPY Penny Pilot	224.1.2.8	11031	224.1.2.40	12031
253	Series Status	A-ZZ	224.1.2.29	11253	224.1.2.61	12253
254	Auction Imbalance	A-ZZ	224.1.2.30	11254	224.1.2.62	12254
255	Last Sales	A-ZZ	224.1.2.31	11255	224.1.2.63	12255

*AAPL is included in subscriptions 3 and 7, and not in subscriptions 0 or 4.

**GOOG is included in subscriptions 24 and 28, and not in subscriptions 9 or 13.

> **Note:** The number of subscriptions and their configurations are not final. This information is subject to change.

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.

In the event 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, Archipelago provides a TCP/IP Recovery Server from which clients can request dropped packets. See the Recovery section for more information.

TCP/IP Recovery

Subscribers may connect to the TCP/IP Recovery Server to request dropped packets from the ArcaBook for Options 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.

Archipelago supplies subscribers with the following parameters:

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

Subscribers supply Archipelago with the IP address for their connection.

3 Messages

Data Types

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

Sequence Numbers

Sequence Numbers for packets and for messages are four byte integers. These numbers start the data feed session at one and increment by one for each new packet or message. See the Recovery section for more information on sequence numbers.

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.

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. 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.
```

Packets

All Packets are encapsulated in variable length Transmission Blocks, as shown below.

Packet Length	**Type**	**Subscription**	**Packet Sequence Number**	**Compacted Messages**

The full length of the packet as 2 byte Numeric Binary	A 1 byte Alpha/Numeric code:
	M = Message
	B = Heartbeat
	N = Not found (for TCP/IP packet replay only)
		The subscription number for the packet as 1 byte Numeric Binary (0 -255)	A 4 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.

Compaction

Archipelago compacts all ArcaBook for Options messages using the FIX Adapted for Streaming (FAST) Protocol. The FAST protocol uses several encoding methods to decrease the size of messages sent over the network. The compaction algorithm reduces the size of messages by an average of 75%. See the FIX FAST Protocol section for a detailed explanation and encoding methods.

Compacted Messages Expanded

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

Message Header and Body	Message Header and Body	Message Header and Body	…

Recovery

In the event a packet is lost on the primary ArcaBook for Options 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. 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

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

ArcaBook for Options Message Header	Offset	Len	Type	Notes and Values
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

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 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 section 5 for information on recovering missing packets.

Aggregate Quote Message	Offset	Len	Type	Notes and Values
Header (8 bytes)				
Message Length	0	2	Binary	52
Message Type	2	1	Alpha/Numeric	‘q’
Subscription	3	1	Binary	0-255 identifying the Subscription
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body(44 Bytes)				

Aggregate Quote Message	Offset	Len	Type	Notes and Values
Message Sequence Number	8	4	Binary	Archipelago assigned sequence number. 1 – 4,294,967,294 within a series.
Strike Price	12	4	Binary	Null (reserved for explicit symbology). Strike price of the series. 4 decimal places.
Expiry Year	16	2	Binary	Null (reserved for explicit symbology). Series expiration year in YYYY format.
Expiry Month	18	1	Binary	Null (reserved for explicit symbology). Series expiration month in MM format.
Expiry Day	19	1	Binary	Null (reserved for explicit symbology). Series expiration day in DD format.
Underlying Quantity	20	2	Binary	Null (reserved for explicit symbology). Shares of the underlying stock in one option contract (1 – 65535)
Put or Call	22	1	Alpha	Null (reserved for explicit symbology). 0=Put 1=Call
Corporate Action	23	1	Alpha	Null (reserved for explicit symbology). A flag indicating this series has been created because of corporate changes in the underlying stock.
Underlying Symbol	24	8	Alpha	Null (reserved for explicit symbology. This field may be populated for internal use until explicit symbology is specified). Underlying stock for this series.
Option Symbol	32	8	Alpha	The full OPRA symbol for this series.
Padding	40	1	Binary	Null

Side	41	1	Alpha	B = Bid
				S = Offer

Insert Price Level	42	1	Binary	New price-level index for this quote (1-5).
Delete Price Level	43	1	Binary	Price-level index to delete for this quote (1-5).
Insert Contracts	44	4	Binary	Aggregated volume of the quote
Insert Price	48	4	Binary	The price for this quote level. 4 decimal places.

Depth of Book Messaging

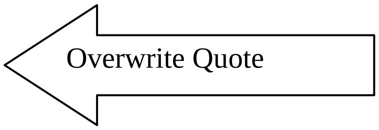
Scenario 1: Update to a Single Price-Level

In this scenario, the price-level indices are the same for insert and delete, indicating that this updates that 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			1000
Depth of Book	1	10	
	2	8	1000
	3	7	500
	4	4	3000
	5	2	400



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

Shift Prices Up

	Price Level	Price	Volume
Modified Depth of Book	1	2	1000
	2	6	200
	3	8	3000
	4	**9**	**500**
	5	10	400

Insert New Price

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

Shift Prices Down

	Price Level	Price	Volume
Modified Depth of Book	1	10	1000
	2	**9**	**500**
	3	8	1000
	4	6	200
	5	2	400

Insert New Price

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
	4	0	0
	5	0	0

Shift Prices Up

	Price Level	Price	Volume
Modified Depth of Book	1	10	1000
	2	6	200
	3	**0**	**0**
	4	0	0
	5	0	0

Insert Zeroed Quote

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**

Shift Prices Down

			1000
Modified Depth of Book	1	10	
	2	**8**	**1000**
	3	6	200
	4	0	0
	5	0	0

Insert New Price

Top of Book Messaging

Scenario 6: 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	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**

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

	Delete Price Level	Insert Price Level	Insert Price	Volume
Original Top of Book		**1**	**6**	**200**
Modified Top of Book		**1**	**0**	**0**

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 5:30 a.m. EST and the conclusion of the Opening Auction. ArcaBook for Options also disseminates imbalance information for Halt Auctions.

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.

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.

Auction Imbalance Message	Offset	Len	Type	Notes and Values
Header (8 bytes)				
Message Length	0	2	Binary	60
Message Type	2	1	Alpha/Numeric	'i'
Subscription	3	1	Binary	0-255 identifying the Subscription
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body(52 Bytes)				
Message Sequence Number	8	4	Binary	Archipelago assigned sequence number. 1 – 4,294,967,294 within a series.
Strike Price	12	4	Binary	Null (reserved for explicit symbology). Strike Price of the series. 4 Decimals.
Expiry Year	16	2	Binary	Null (reserved for explicit symbology). Series expiration year in YYYY format.
Expiry Month	18	1	Binary	Null (reserved for explicit symbology). Series expiration month in MM format.
Expiry Day	19	1	Binary	Null (reserved for explicit symbology). Series expiration day in DD format.

Auction Imbalance Message	Offset	Len	Type	Notes and Values
Underlying Quantity	20	2	Binary	Null (reserved for explicit symbology). Shares of the underlying stock in on option contract (1 – 65535)
Put or Call	22	1	Alpha	Null (reserved for explicit symbology). 0=Put 1=Call
Corporate Action	23	1	Alpha	Null (reserved for explicit symbology). A flag indicating this series has been created because of corporate changes in the underlying stock.
Underlying Symbol	24	8	Alpha	Null (reserved for explicit symbology. This field may be populated for internal use until explicit symbology is specified). Underlying stock for this series.
Option Symbol	32	8	Alpha	The full OPRA symbol for this series.
Padding	40	1	Binary	Null

Auction Type	41	1	Alpha	O = Open,
				M = Market
				H = Halt
				C = Closing

Auction Time	42	2	Binary	Projected auction time (hhmm) 0 – 2400
Contracts	44	4	Binary	Indicative match volume.
Price	48	4	Binary	The indicative match price
Total Imbalance	52	4	Binary	The total imbalance volume
Market Imbalance	56	4	Binary	The market imbalance volume

System Event Message

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

System Event Message	Offset	Len	Type	Notes and Values
Header (8 bytes)				
Message Length	0	2	Binary	44
Message Type	2	1	Alpha/Numeric	‘v’
Subscription	3	1	Binary	0-255 identifying the Subscription
Time Stamp	4	4	Binary	Milliseconds since Midnight
Message Body(32 Bytes)				

System Event Message	Offset	Len	Type	Notes and Values
Message Sequence Number	8	4	Binary	Archipelago assigned sequence number. 1 – 4,294,967,294 within a series. 0 = if the message clears quotes by Underlying or by Subscription
Strike Price	12	4	Binary	Null (reserved for explicit symbology). Strike Price of the series. 0 = if the message clears quotes by Underlying or by Subscription
Expiry Year	16	2	Binary	Null (reserved for explicit symbology). Expiration year in YYYY format. 0 = if the message clears quotes by Underlying or by Subscription
Expiry Month	18	1	Binary	Null (reserved for explicit symbology). Expiration month in MM format. 0 = if the message clears quotes by Underlying or by Subscription
Expiry Day	19	1	Binary	Null (reserved for explicit symbology). Expiration day in DD format. 0 = if the message clears quotes by Underlying or by Subscription
Underlying Quantity	20	2	Binary	Null (reserved for explicit symbology). Shares of the underlying stock in one option contract (1 – 65535) 0 = if the message clears quotes by Underlying or by Subscription
Put or Call	22	1	Alpha	Null (reserved for explicit symbology). 0 (ASCII)= Put 1 = Call 0 (numeric) = if the message clears quotes by Underlying or by Subscription
Corporate Action	23	1	Alpha	Null (reserved for explicit symbology). A flag indicating this series has been created because of corporate changes in the underlying stock. 0 (numeric) = if the message clears quotes by Underlying or by Subscription
Underlying Symbol	24	8	Alpha	Null (reserved for explicit symbology. This field may be populated for internal use until explicit symbology is specified). Underlying stock for this series. 0 = if the message clears quotes by Subscription
Option Symbol	32	8	Alpha	The full OPRA symbol for this series.
Padding	40	2	Binary	Null

System Event Message	Offset	Len	Type	Notes and Values

Event Code	42	1	Alpha	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
				L = Light up a dark series
				N = Open Indication (Dark series)
				O = Open Indication (Series)
				o = Open Indication (Underlying)
				S = Suspend(Halt Series)
				T = Unhalt Dark Series
				U = Unhalt Series.
				s = Suspend(Halt Underlying)
				u = Unhalt Underlying.
				X = Close Indication (Series)
				x = Close Indication (Underlying)
				** reserved for explicit symbology
				*Dark series is a series won't be quoted if there haven't been any trades in that Series for specific number of days

Reset Code	43	1	Alpha	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'.

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 ArcaBook users. Please see the list below for Series Status message frequency.

- Stock Opens - 1 Message gets sent, covers for all series
- Stock Closes - 1 Message gets sent, covers for the stock
- Stock Halts - 1 Message gets sent, covers for all series
- Stock unhalts - 1 Message gets sent, covers for all series
- Series Opens - A Message get sent for each **individual** series

- Series Closes - 1 Message gets sent, covers for all series
- Series Halts - 1 Message gets sent, covers for all series
- Series UnHalts - A Message gets sent for each **individual** series

5 TCP/IP Recovery Messages

ArcaBook for Options Recovery Server Messages

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.

HeartBeat Message	Offset	Len	Type	Notes and Values
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

Test Response Message

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

Test Response Message	Offset	Len	Type	Notes and Values
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

Login Accepted Message

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

Login Accepted Message	Offset	Len	Type	Notes and Values
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	

Login Rejected Message

The ArcaBook for Options 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. ArcaBook for Options closes the socket connection after sending this message.

Login Rejected Message	Offset	Len	Type	Notes and Values
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	

Packet Replay Message

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

Packet Replay Message	Offset	Len	Type	Notes and Values
Header (8 bytes)				
Packet Length	0	2	Binary	Size of Compacted Messages
Message Type	2	1	Alpha	See the Packets section 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	8	Variable	FIX FAST Compacted	See the UDP Multicast Messages section for more information.

Messages Sent by a Subscriber

Login Request Message

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

Logon Request Message	Offset	Len	Type	Notes and Values
Header (8bytes)				
Message Length	0	2	Binary	28 bytes
Message Type	2	1	Alpha/Numeric	‘L’
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

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.

HeartBeat Response Message	Offset	Len	Type	Notes and Values
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)

Test Request Message

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

Test Request Message	Offset	Len	Type	Notes and Values
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 (not required)
Message Body(20 Bytes)				
Test Message	8	20	Alpha	Text to be echoed

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.

Packet Request Message	Offset	Len	Type	Notes and Values
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
Ending Packet Number	12	4	Binary	1 – 4,294,967,294

6 FIX FAST Protocol

Overview

Subscribers receive the 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 and reduced latency. 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 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.

  The combination of field encoding templates and presence maps allows the contents of a message to be communicated fully while reducing the number of bytes on the wire.
- **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.

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.

The ArcaBook for Options FAST Implementation

The ArcaBook for Options FAST implementation reduces bandwidth requirements by up to 80%, or a ratio of 5 to 1. Each message within the FAST 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.

Sample Source Code

To help subscribers process the ArcaBook for Options FAST feed, Archipelago provides source code to decode ArcaBook for Options FAST messages into ArcaBook for Options binary messages. A single, C language routine, AQFastDecode() , decodes 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.

Template Information

The FAST template for each message indicates which fields may be omitted from a message and how clients should interpret omitted fields. 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 1.

Table 1: ArcaBook for Options FAST Message Template

Field ID	Field Name	In Messages of Type	FAST Type	Encoding
0	AB_MSG_TYPE	All	Unsigned8	None
1	AB_SUBSCRIPTION	All	Unsigned8	Copy
2	AB_TIME	All	Unsigned32	Copy
3	AB_SEQUENCE	q, i and v	Unsigned32	Increment
4	AB_STRIKE_PRICE	q, i and v	Unsigned32	Copy
5	AB_EXPIRY_YEAR	q, i and v	Unsigned16	Copy
6	AB_EXPIRY_MONTH	q, i and v	Unsigned8	Copy
7	AB_EXPIRY_DAY	q, i and v	Unsigned8	Copy
8	AB_UNDERLYING_QUANTITY	q, i and v	Unsigned16	Copy
9	AB_PUT_CALL	q, i and v	Unsigned8	Copy
10	AB_CORP_ACTION	q, i and v	Unsigned8	Copy
11	AB_UNDERLYING_SYMBOL	q, i and v	Unsigned8	Copy
12	AB_OPTION_SYMBOL	q, i and v	Unsigned8	Copy
13	AB_SIDE	q	Unsigned8	Copy
	AB_AUCTION_TYPE	i		
	AB_EVENT_CODE	v		
14	AB_INSERT_PRICE_LEVEL	q	Unsigned8	Copy
	AB_RESET_CODE	v		
15	AB_DELETE_PRICE_LEVEL	q	Unsigned8	Copy
16	AB_INSERT_CONTRACTS	q	Unsigned32	Copy
	AB_CONTRACTS	i		
17	AB_INSERT_PRICE	q	Unsigned32	Copy
	AB_INDICATIVE_PRICE	i		
18	AB_TOTAL_IMBALANCE	i	Unsigned32	Copy
19	AB_MKT_IMBALANCE	i	Unsigned32	Copy
20	AB_AUCTION_TIME	I	Unsigned16	Copy
	AB_BITMAP	Any **except** q, i and v	Bitmap	None

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