

ArcaBook® Specification for Options

For NYSE Arca and AMEX Options Exchanges

*January 31, 2009
Version 3.05*

Copyright © 2009 Archipelago Holdings, Inc. All Rights Reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written permission of Archipelago Holdings, Inc. NYSE Arca®, Amex®, ArcaEx®, ArcaBook® and ARCA® are registered trademarks of NYSE Group, Inc. or its affiliates. Third party product names used herein are used to identify such products and for descriptive purposes only. Such names may be marks and/or registered marks of their respective owners.

Archipelago Holdings, Inc. and NYSE Group, Inc. are a wholly owned subsidiaries of NYSE Euronext.

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)
10-17-08	3.0	Chris McCown	New Underlying and Series Index Message Index Mapping Request. New Message Formats, New IPs.
11-14-2008	3.01	Chris McCown	Included Old IPs
12-12-2008	3.02	David Le	Update Fast Template
01-14-2009	3.03	David Le	Added source IP
01-22-2009	3.04	Anil Nagasamudra	Added additional details to index mapping request. Made underlying quantity reserved for future use.

Table of Contents

1 Introduction	6
ArcaBook Interface for Options	6
Data Feeds	6
Depth of Book Messaging	6
Archipelago API Certification	7
2 Communication	8
Access	8
Multicast Quote Feeds	8
TCP/IP Recovery	11
3 Messages	12
Data Types	12
Sequence Numbers	12
Prices	12
Timestamps	12
Packets	12
Compaction	13
Compacted Messages Expanded	13
Recovery	13
4 UDP Multicast Messages	14
Underlying Index Mapping	14
Series Index Mapping	15
Aggregate Quote Message	16
Depth of Book Messaging	17
Top of Book Messaging	21
Imbalance Message	22
Market Order Imbalance	22
Total Imbalance	22
System Event Message	23
5 TCP/IP Recovery Messages	26
ArcaBook for Options Recovery Server Messages	26
HeartBeat Request Message	26
Test Response Message	26
Login Accepted Message	26
Login Rejected Message	27
Packet Replay Message	27
Messages Sent by a Subscriber	27
Login Request Message	27
Heartbeat Response Message	28
Index Mapping Request Message	28
Test Request Message	28
Dropped Packet Request Message	29

6 FIX FAST Protocol	30
Overview	30
A FAST Message	30
The ArcaBook for Options FAST Implementation.....	31
Sample Source Code.....	31
Template Information	Error! Bookmark not defined.

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 (Option 1) or email connectivity@nyx.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.

Subscription	Type	Underlyings	NYSE Arca Primary Multicast IP	NYSE Arca Secondary Multicast IP	AMEX Primary Multicast IP	AMEX Secondary Multicast IP	Primary Port	Secondary Port
0	Depth	A	224.0.41.0	224.0.41.128	224.0.58.0	224.0.58.128	11000	12128
1	Depth	B	224.0.41.1	224.0.41.129	224.0.58.1	224.0.58.129	11001	12129
2	Depth	C	224.0.41.2	224.0.41.130	224.0.58.2	224.0.58.130	11002	12130
3	Depth	D	224.0.41.3	224.0.41.131	224.0.58.3	224.0.58.131	11003	12131
4	Depth	E	224.0.41.4	224.0.41.132	224.0.58.4	224.0.58.132	11004	12132
5	Depth	F	224.0.41.5	224.0.41.133	224.0.58.5	224.0.58.133	11005	12133
6	Depth	G	224.0.41.6	224.0.41.134	224.0.58.6	224.0.58.134	11006	12134
7	Depth	H	224.0.41.7	224.0.41.135	224.0.58.7	224.0.58.135	11007	12135
8	Depth	I	224.0.41.8	224.0.41.136	224.0.58.8	224.0.58.136	11008	12136
9	Depth	J	224.0.41.9	224.0.41.137	224.0.58.9	224.0.58.137	11009	12137
10	Depth	K	224.0.41.10	224.0.41.138	224.0.58.10	224.0.58.138	11010	12138
11	Depth	L	224.0.41.11	224.0.41.139	224.0.58.11	224.0.58.139	11011	12139
12	Depth	M	224.0.41.12	224.0.41.140	224.0.58.12	224.0.58.140	11012	12140
13	Depth	N	224.0.41.13	224.0.41.141	224.0.58.13	224.0.58.141	11013	12141
14	Depth	O	224.0.41.14	224.0.41.142	224.0.58.14	224.0.58.142	11014	12142
15	Depth	P	224.0.41.15	224.0.41.143	224.0.58.15	224.0.58.143	11015	12143
16	Depth	Q	224.0.41.16	224.0.41.144	224.0.58.16	224.0.58.144	11016	12144
17	Depth	R	224.0.41.17	224.0.41.145	224.0.58.17	224.0.58.145	11017	12145
18	Depth	S	224.0.41.18	224.0.41.146	224.0.58.18	224.0.58.146	11018	12146
19	Depth	T	224.0.41.19	224.0.41.147	224.0.58.19	224.0.58.147	11019	12147
20	Depth	U	224.0.41.20	224.0.41.148	224.0.58.20	224.0.58.148	11020	12148
21	Depth	V	224.0.41.21	224.0.41.149	224.0.58.21	224.0.58.149	11021	12149

22	Depth	W	224.0.41.22	224.0.41.150	224.0.58.22	224.0.58.150	11022	12150
23	Depth	X	224.0.41.23	224.0.41.151	224.0.58.23	224.0.58.151	11023	12151
24	Depth	Y	224.0.41.24	224.0.41.152	224.0.58.24	224.0.58.152	11024	12152
26	Depth	Z	224.0.41.25	224.0.41.153	224.0.58.25	224.0.58.153	11025	12153
32	Top	A	224.0.41.32	224.0.41.160	224.0.58.32	224.0.58.160	11032	12160
33	Top	B	224.0.41.33	224.0.41.161	224.0.58.33	224.0.58.161	11033	12161
34	Top	C	224.0.41.34	224.0.41.162	224.0.58.34	224.0.58.162	11034	12162
35	Top	D	224.0.41.35	224.0.41.163	224.0.58.35	224.0.58.163	11035	12163
36	Top	E	224.0.41.36	224.0.41.164	224.0.58.36	224.0.58.164	11036	12164
37	Top	F	224.0.41.37	224.0.41.165	224.0.58.37	224.0.58.165	11037	12165
38	Top	G	224.0.41.38	224.0.41.166	224.0.58.38	224.0.58.166	11038	12166
39	Top	H	224.0.41.39	224.0.41.167	224.0.58.39	224.0.58.167	11039	12167
40	Top	I	224.0.41.40	224.0.41.168	224.0.58.40	224.0.58.168	11040	12168
41	Top	J	224.0.41.41	224.0.41.169	224.0.58.41	224.0.58.169	11041	12169
42	Top	K	224.0.41.42	224.0.41.170	224.0.58.42	224.0.58.170	11042	12170
43	Top	L	224.0.41.43	224.0.41.171	224.0.58.43	224.0.58.171	11043	12171
44	Top	M	224.0.41.44	224.0.41.172	224.0.58.44	224.0.58.172	11044	12172
45	Top	N	224.0.41.45	224.0.41.173	224.0.58.45	224.0.58.173	11045	12173
46	Top	O	224.0.41.46	224.0.41.174	224.0.58.46	224.0.58.174	11046	12174
47	Top	P	224.0.41.47	224.0.41.175	224.0.58.47	224.0.58.175	11047	12175
48	Top	Q	224.0.41.48	224.0.41.176	224.0.58.48	224.0.58.176	11048	12176
49	Top	R	224.0.41.49	224.0.41.177	224.0.58.49	224.0.58.177	11049	12177
50	Top	S	224.0.41.50	224.0.41.178	224.0.58.50	224.0.58.178	11050	12178
51	Top	T	224.0.41.51	224.0.41.179	224.0.58.51	224.0.58.179	11051	12179
52	Top	U	224.0.41.52	224.0.41.180	224.0.58.52	224.0.58.180	11052	12180
53	Top	V	224.0.41.53	224.0.41.181	224.0.58.53	224.0.58.181	11053	12181
54	Top	W	224.0.41.54	224.0.41.182	224.0.58.54	224.0.58.182	11054	12182
55	Top	X	224.0.41.55	224.0.41.183	224.0.58.55	224.0.58.183	11055	12183
56	Top	Y	224.0.41.56	224.0.41.184	224.0.58.56	224.0.58.184	11056	12184
57	Top	Z	224.0.41.57	224.0.41.185	224.0.58.57	224.0.58.185	11057	12185
124	Complex	A-Z	224.0.41.59	224.0.41.187	224.0.58.59	224.0.58.187	11059	12187
125	Series Status	A-Z	224.0.41.61	224.0.41.189	224.0.58.61	224.0.58.189	11061	12189
126	Auction Imbalance	A-Z	224.0.41.62	224.0.41.190	224.0.58.61	224.0.58.190	11062	12190

127	Last Sales	A-Z	224.0.41.63	224.0.41.191	224.0.58.63	224.0.58.191	1163	12191
-----	------------	-----	-------------	--------------	-------------	--------------	------	-------

Note: The number of subscriptions and their configuration has not been finalized. 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 Source IP addresses

The table below outlines the TCP Source IP addresses, applicable to each Port.

TCP Source IP NYSE ARCA	TCP Source IP AMEX
63.211.72.xxx	208.92.194.xxx
8.9.19.xxx	8.9.33.xxx

Current OX CERT Breakout

Subscription	Type	Underlyings	Primary Multicast IP	Primary Port	Secondary Multicast IP	Secondary Port
0	Depth	A-CSCO	224.1.2.27	10010	224.1.2.47	20010
1	Depth	CSE – MSFT	224.1.2.27	10011	224.1.2.47	20011
2	Depth	MSM - ZZ	224.1.2.27	10012	224.1.2.47	20012
3	Depth	Penny Pilots	224.1.2.27	10013	224.1.2.47	20013
4	Top	AA - ARE	224.1.2.27	10014	224.1.2.47	20014
5	Top	ARG - BRNC	224.1.2.27	10015	224.1.2.47	20015
6	Top	BRO - CNX	224.1.2.27	10016	224.1.2.47	20016

7	Top	Penny Pilots	224.1.2.27	10017	224.1.2.47	20017
252	Complex	A-ZZ	224.1.2.27	10080	224.1.2.47	20080
253	Series Status	A-ZZ	224.1.2.27	10070	224.1.2.47	20070
254	Auction Imbalance	A-ZZ	224.1.2.27	10060	224.1.2.47	20060
255	Last Sales	A-ZZ	224.1.2.27	10050	224.1.2.47	20050

TCP Source IP CERT addresses

The table below outlines the TCP Source IP addresses, applicable to each Port.

TCP Source IP CERT NYSE ARCA
63.211.72.xxx

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 $135000 \div 10,000 = 13.50$.
- **Example 2:** Whole integer price is 13500. The decimal price is $13500 \div 10,000 = 1.35$.

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:

$\text{Seconds} \times 1000 + \text{milliseconds}$

For example, the timestamp for 10:00:00:376 is converted to:

$(36000 \times 1000) + 376 = 36000376$.

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 uncompact, 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

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.

Underlying Index Mapping Message	Offset	Len	Type	Fast Template	Notes and Values
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

Underlying Index Mapping Message	Offset	Len	Type	Fast Template	Notes and Values
					'A' = AMEX
Security Type	23	1	Alpha	AQ_SECURITY_TYPE	A = ADR C = Common E = ETF F = Foreign, 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 Padded
Reserved3	30	1	Alpha	AQ_RESERVE_3	NULL
Reserved4	31	1	Alpha	AQ_RESERVE_4	NULL

Series Index Mapping

Series Index Messages will be immediately follow the underlying index messages. Each subscription will send the series mappings used on it's subscription.

Series Index Mapping Message	Offset	Len	Type	Fast Template	Notes and Values
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 identifier
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since epoch
Message Body(52 Bytes)					
Series Index	8	4	Binary	AQ_SERIES_INDEX	Series Mapping ID
Market ID	12	2	Binary	AQ_MARKET_ID	Identifies Market
System ID	14	1	Binary	AQ_SYSTEM_ID	Identifies Trad
Bit	15	1	Alpha	AQ_BIT	NULL
Underlying Index	16	4	Binary	AQ_UNDERLYING_INDEX	Underlying Sto
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 F

Series Index Mapping Message	Offset	Len	Type	Fast Template	Notes and Values
Underlying Symbol	32	6	Alpha	AQ_SYMBOL	Underlying Symbol 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
Strike Price Decimal	50	3	Alpha	AQ_STRIKE_DECIMAL	Left Justified, 0
Price Scale	53	1	Binary	AQ_PRICE_SCALE	Decimal place
Option Symbol	54	5	Alpha	AQ_OCC_SYMBOL	The full OPRA NULL padded.
Reserved4	59	1	Alpha	AQ_RESERVE_4	NULL

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	Fast Template	Notes and Values
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 th
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since 1
Message Body(32 Bytes)					

Aggregate Quote Message	Offset	Len	Type	Fast Template	Notes and Values
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 starting at 4,294,967,294 with an interval of 100,000,000.
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
Price	32	4	Binary	AQ_PRICE	The price at the time of the quote
Delete Price Level	36	1	Binary	AQ_DELETE_LEVEL	Price-level index to be deleted (1-5).
Insert Price Level	37	1	Binary	AQ_INSERT_LEVEL	New price-level index to be added (1-5).
Side	38	1	Alpha	AQ_BUY_SELL	B = Bid S = Offer
Price Scale	39	1	Binary	AQ_PRICE_SCALE	Null (Not Use)

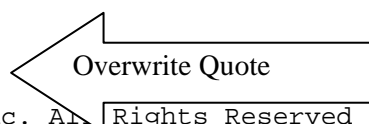
Depth of Book Messaging

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 Depth of Book			1000
	1	10	
	2	8	1000



Price Level	Price	Volume
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

Original Depth of Book	Price Level	Price	Volume
	1	2	1000
	2	4	1000
	3	6	200
	4	8	3000
	5	10	400



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

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	

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

Scenario 4: Delete on a partial depth book.

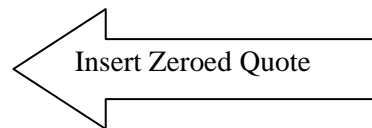
In this scenario, the depth of the book is less than 5 and a Price Level must be removed from the book.

	Delete Price Level	Insert Price Level	Insert Price	Volume
Bid Quote	2	3	0	0

Original Depth of Book	Price Level	Price	Volume
	1	10	1000
	2	8	1000
	3	6	200
	4	0	0
	5	0	0



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



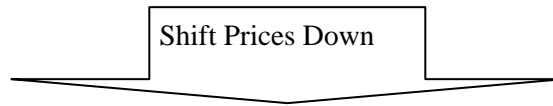
Scenario 5: Insert on a partial depth book.

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

Original Depth of Book	Price Level	Price	Volume
	1	10	1000
	2	6	200
	3	0	0

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



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	Fast Template	Notes and Values
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 – 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)

Auction Imbalance Message	Offset	Len	Type	Fast Template	Notes and Values
					2400
Auction Type	38	1	Alpha	AQ_AUCTION_TYPE	O = Open, M = Market H = Halt C = Closing
Reserved4	39	1		AQ_RESERVE_4	NULL

System Event Message

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

System Event Message	Offset	Len	Type	Fast Template	Notes and Values
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
Time Stamp	4	4	Binary	AQ_TIME	Milliseconds since Mi
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	Archipelago assigned number. 1 – 4,294,96 series. 0 = if the message cleared Underlying or by Sub
Reserved5	20	2	Alpha	AQ_RESERVE_5	NULL

System Event Message	Offset	Len	Type	Fast Template	Notes and Values
Event Code	22	1	Alpha	AQ_EVENT_CODE	A = Clear Series Ask B = Clear Series Bid C = Clear Series D = Clear Underlying E = Clear Underlying F = Clear Underlying G = Clear Subscription H = Clear Subscription I = Clear Subscription L = Light up a dark series N = Open Indication (N) O = Open Indication (O) o = Open Indication (o) S = Suspend(Halt Series) T = Unhalt Dark Series U = Unhalt Series. s = Suspend(Halt Underlying) u = Unhalt Underlying X = Close Indication (X) x = Close Indication (x) ** reserved for explicit *Dark series is a series that is not active if there haven't been any trades in the series Series for specific numbers.
Reset Code	23	1	Alpha	AQ_RESET_CODE	C = Continue using expected series numbers. R = Reset <i>Packet Sequence</i> all expected <i>Message Numbers</i> <i>Numbers</i> to 1 for this message will be in a Number of 1. This will be the first message of the day, if the system is up. This message will also be sent during System Failover. This message is sent in conjunction with an Unhalt message.

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)

Index Mapping Request Message

Clients send this message to request a response from the ArcaBook for Options 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(125). The mapping messages are sent over the TCP socket uncompact for all subscriptions.

Index Mapping Request Message	Offset	Len	Type	Notes and Values
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)

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 back in 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	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,		
	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

Field ID	Field Name	In Messages of Type	FAST Type	Encoding
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 ID's with multiple Field Names are guaranteed never to occur more than once in a given message.