BATS US Equities

Binary Order Entry Specification

Version 2.1.2

July 5, 2015

Contents

1 Introduction

1.1 Overview

This document describes BATS Binary Order Entry (BOE), the BATS Global Markets proprietary order entry protocol.

Where applicable, the terminology (e.g., time in force) used in this document is similar to that used by the FIX protocol to allow those familiar with FIX to more easily understand BOE. This document assumes the reader has basic knowledge of the FIX protocol.

BOE fulfills the following requirements:

- *CPU and memory efficiency.* Message encoding, decoding, and parsing are simpler to code and can be optimized to use less CPU and memory at runtime.
- *Application level simplicity.* State transitions are simple and unambiguous. They are easy to apply to a Member's representation of an order.
- *Session level simplicity.* The session level protocol (login, sequencing, replay of missed messages, logout) is simple to understand.

Whilst BATS has strived to preserve feature parity between FIX and BOE where possible, some features may only be available in one protocol or the other.

All binary values are in little Endian (used by Intel x86 processors), and *not* network byte order.

Each message is identified by a unique message type. Not all message types are used in all of BATS' trading environments globally. A listing of the supported message types is provided in **List of Message Types** (§ 9, p. 69).

All communication is via standard TCP/IP.

1.2 Motivation for Version 2

BOE Version 1 has a number of fixed size parts of messages which, while envisioned to be large enough for future growth, have been unable to accomodate BATS' growth into new service offerings. Version 2 allows greater opportunity for future expansion by eliminating those problems.

Version 2's goals are as follows:

- *Return bitfield expansion.* Messages from BATS to Member no longer have a limited number of return bitfields. Members may ignore newly added fields as before, but there is no longer a fixed limit to the number of possible fields returned.
- *Login message parameter groups.* In Version 2, the Login Request V2 message can have extendable parameter groups sent to modify behavior in a forward compatible manner.
- *Easy extension of messages from Member to BATS to support more bitfields.* In Version 1, messages such as New Order supported a fixed number of bitfields. In Version 2, New Order V2 requires that the number of entered bitfields be specified. This supports, in a backwards compatible way, addition of new bitfields in the future.
- *Easier addition of new messages.* In Version 1, the return bitfields for *all* messages had to be represented in the Login Request. Addition of messages meant changes to the fundamental structure of the Login Request. In Version 2, repeatable parameter groups are used to specify which bitfields are to be sent for different message type. This allows the Login Request V2 to accomodate new message types without fundamental changes to the message structure.
- *Latency improvements.* As part of the move to Version 2, BATS is taking the opportunity to optimize the message encoding (BATS to Member) to further reduce latency on return messages.

- *Simplification of documentation.* BATS has reduced the complexity of this documentation to make BOE easier to understand.

If you are newly developing to the BATS BOE, you should implement to Version 2 of the specification. Newly added features (e.g., new message fields) *may* be implemented only in Version 2. You may migrate to Version 2 at any point, but you will be *required* to migrate to Version 2 if and when you require use of such features.

To the extent possible, Version 2 has a similar "look and feel" to Version 1. Session-level concepts such as sequencing and heartbeats are identical. Only messages documented in Version 2 are supported on a connection established with a LOGIN REQUEST V2. Data type encoding remains identical. A design goal for the evolution to Version 2 was to make it possible to upgrade Version 1 code to support Version 2 with a minimal amount of development effort.

1.3 Data Types

The following data types are used by BOE. The size of some data types varies by message. All data types have default values of binary zero, in both Member to BATS and BATS to Member contexts.

- *Binary*: Little Endian byte order, unsigned binary value. The number of bytes used depends on the context.
  - One byte: `FE` = 254
  - Four bytes: `64 00 00 00` = 100
- *Signed Binary*: Little Endian byte order, signed two's complement, binary value. The number of bytes used depends on the context.
  - One byte: `DF` = −33
  - Four bytes: `64 00 00 00` = +100
- *Binary Price*: Little Endian byte order value, eight bytes in size, with four implied decimal places. So, if the value is 123,400, the actual value taking into account implied decimal places is 12.34.
  - `08 E2 01 00 00 00 00 00` = 123,400/10000 = 12.34
- *Signed Binary Price*: Little Endian byte order value, signed two's complement, eight bytes in size, with four implied decimal places. So, if the value is −123,400, the actual value taking into account implied decimal places is −12.34.
  - `08 E2 01 00 00 00 00 00` = 123,400/10000 = 12.34
  - `F8 1D FE FF FF FF FF FF` = −123,400/10000 = −12.34
- *Short Binary Price*: Little Endian byte order value, four bytes in size, with four implied decimal places. So, if the value is 12,300, the actual value taking into account implied decimal places is 1.23.
  - `0C 30 00 00` = 12,300/10000 = 1.23
- *Signed Binary Fee*: Little Endian byte order value, eight bytes in size, signed, with five implied decimal places. So, the value −123,000 is −1.23 after taking account for the five implied decimal places.
  - `88 1F FE FF FF FF FF FF` = −123,000/100000 = −1.23
- *Alpha*: Uppercase letters (A–Z) and lowercase letters (a–z) only. ASCII NUL (0x00) filled on the right, if necessary. The number of bytes used depends on the context.

- *Alphanumeric*: Uppercase letters (A–Z), lowercase letters (a–z) and numbers (0–9) only. ASCII NUL (0x00) filled on the right, if necessary.
- *Text*: Printable ASCII characters only. ASCII NUL (0x00) filled on the right, if necessary.
- *DateTime*: 8 bytes. The date and time, in UTC, represented as nanoseconds past the UNIX epoch (00:00:00 UTC on 1 January 1970). The nanoseconds portion is currently ignored and treated as 0 (i.e. the times are only accurate to microseconds) on input, and will always be set to 0 by BATS in outgoing messages. However, BATS may begin populating the nanoseconds portion at any time without warning.

  For example: 1,294,909,373,757,324,000 = `2011-01-13 09:02:53.757324 UTC`.

1.4 Optional Fields and Bitfields

Some messages such as New Order V2 and Modify Order V2 have a number of optional fields. A count and number of bitfields in the message specify which optional fields will be present at the end of the message. If a bit is set, the field will be present. Fields are appended to the end of the message. There is no implicit framing between the optional fields. In order to decode the optional fields, they *must* be appended in a particular order to the end of the message. The fields of the first bitfield are appended first, lowest order bit first. Next, the fields of the next bitfield are appended, lowest order bit first. This continues for all bitfields. While certain *reserved* bits within a defined bitfield are used within another BATS market and will be ignored, bits that are reserved for future expansion must be set to 0 when noted in the bitfield description.

The size, data type, and values for each field are described in **List of Optional Fields** (§ 7, p. 57).

Note that the set of optional fields returned for each BATS to Member message type is determined at session login (using the Login Request V2 message); hence, the exact size and layout of each message received by the client application can be known in advance. Any requested optional field which is irrelevant in a particular context will still be present in the returned message, but with all bytes set to binary zero (0x00).

Each return message from BATS to Member indicates the optional fields which are present, even though the Member indicated during login which optional fields are to be sent. The reason for the inclusion (and duplication) is so that each message can be interpreted on its own, without having to find the corresponding login request or response to know which optional fields are present. So, for example, in a log file, decoding a message requires only that single message.

Example messages are shown with each message type which should help to make this concept clear.

2 Session

2.1 Message Headers

Each message has a ten byte header. The two initial *StartOfMessage* bytes are present to aid in message reassembly for network capture purposes. The *MatchingUnit* field is only populated on sequenced non-session level messages sent from BATS to the Member. Messages from Member to BATS and all session level messages must always set this value to 0.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	Message type.
MatchingUnit	5	1	Binary	The matching unit which created this message. Matching units in BOE correspond to matching units on Multicast PITCH. **For session level traffic, the unit is set to 0.** **For messages from Member to BATS, the unit must be 0.**
SequenceNumber	6	4	Binary	The sequence number for this message. Messages from BATS to Member are sequenced distinctly per matching unit. Messages from Member to BATS are sequenced across all matching units with a single sequence stream. Member can optionally send a 0 sequence number on all messages from Member to BATS. BATS highly recommends Member to send sequence number on all inbound messages.

2.2 Login, Replay and Sequencing

Session level messages, both inbound (Member to BATS) and outbound (BATS to Member) are unsequenced.

Inbound (Member to BATS) application messages are sequenced. Upon reconnection, BATS informs the Member of the last processed sequence number; the Member *may* choose to resend any messages with sequence numbers greater than this value. A gap forward in the Member's incoming sequence number is permitted at any time and is ignored by BATS. Gaps backward in sequence number (including the same sequence number used twice) are never permitted and will always result in a Logout message being sent and the connection being dropped.

Most (but not all) outbound (BATS to Member) application messages are monotonically sequenced per matching unit. Each message's documentation will indicate whether it is sequenced or unsequenced. While matching units on BOE correspond directly to matching units on Multicast PITCH, sequence numbers do not.

Upon reconnection, a Member sends the last received sequence number per matching unit in a Login Request V2 message. BATS will respond with any missed messages. However, when the Login Request

V2 *NoUnspecifiedUnitReplay* flag is enabled, BATS will exclude messages from unspecified matching units during replay. BATS will send a Replay Complete message when replay is finished. If there are no messages to replay, a Replay Complete message will be sent immediately after a Login Response V2 message. BATS will reject all orders during replay.

Assuming Member has requested replay messages using a properly formatted Login Request V2 after a disconnect, any unacknowledged orders remaining with the Member after the Replay Complete message is received should be assumed to be unknown to BATS.

Unsequenced messages will not be included during replay.

A session is identified by the username and session sub-identifier (both supplied by BATS). Only one concurrent connection per username and session sub-identifier is permitted.

If a login is rejected, an appropriate Login Response V2 message will be sent and the connection will be terminated.

2.3 Sequence Reset

A reset sequence operation is not available for Binary Order Entry. However, a Member can send a Login Request message with *NoUnspecifiedUnitReplay* field enabled, and *NumberOfUnits* field set to zero. Then, upon receiving a Login Response V2 message from BATS, the Member can use the field *LastReceivedSequenceNumber* as the sequence starting point for sending future messages.

2.4 Heartbeats

Client Heartbeat messages are sent from Member to BATS and Server Heartbeat messages are sent from BATS to Member if no other data has been sent in that direction for one second. Like other session level messages, heartbeats from BATS to the Member do *not* increment the sequence number. If BATS receives no inbound data or heartbeats for five seconds, a Logout message will be sent and the connection will be terminated. **Members are encouraged to have a one second heartbeat interval and to perform similar connection staleness logic.**

2.5 Logging Out

To gracefully log out of a session, a Logout Request message should be sent by the Member. BATS will finish sending any queued data for that port and will then respond with its own Logout message and close the connection. After receipt of a Logout Request message, BATS will ignore all other inbound (Member to BATS) messages except for Client Heartbeat.

3 Session Messages

3.1 Member to BATS

3.1.1 Login Request V2

A Login Request V2 message must be sent as the first message upon connection.

A number of repeating parameter groups, some of which may be required, are sent at the end of the message. Ordering of parameter groups is not important. New parameter groups may be added in the future with no notice.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x37`
MatchingUnit	5	1	Binary	Always 0 for inbound (Member to BATS) messages.
SequenceNumber	6	4	Binary	Always 0 for session level messages.
SessionSubID	10	4	Alphanumeric	Session Sub ID supplied by BATS.
Username	14	4	Alphanumeric	Username supplied by BATS.
Password	18	10	Alphanumeric	Password supplied by BATS.
NumberOfParam Groups	28	1	Binary	A number, n (possibly 0), of parameter groups to follow.
$ParamGroup_1$				First parameter group.
⋮				
$ParamGroup_n$				Last parameter group.

Unit Sequences Parameter Group

This parameter group includes the last consumed sequence number per matching unit received by the Member. BATS uses these sequence numbers to determine what outbound (BATS to Member) traffic, if any, was missed by the Member. If this parameter group is not sent, it's assumed the Member has not received any messages (e.g., start of day).

The Member does *not* need to include a sequence number for a unit if they have never received messages from it. For example, if the Member has received responses from units 1, 3, and 4, the Login Request V2 message need not include unit 2. If the Member wishes to send a value for unit 2 anyway, 0 would be the only allowed value.

Only one instance of this parameter group may be included.

Field	Offset	Length	Data Type	Description
ParamGroupLength	0	2	Binary	Number of bytes for the parameter group, including this field.
ParamGroupType	2	1	Binary	`0x80`
NoUnspecified UnitReplay	3	1	Binary	Flag indicating whether to replay missed outgoing (BATS to Member) messages for unspecified units. `0x00` = False (Replay Unspecified Units) `0x01` = True (Suppress Unspecified Units Replay)

NumberOfUnits	4	1	Binary	A number, n (possibly 0), of unit/sequence pairs to follow, one per unit from which the Member has received messages.
$UnitNumber_1$		1	Binary	A unit number.
$UnitSequence_1$		4	Binary	Last received sequence number for the unit.
⋮				
$UnitNumber_n$		1	Binary	A unit number.
$UnitSequence_n$		4	Binary	Last received sequence number for the unit.

Return Bitfields Parameter Group

This parameter group, which may be repeated, indicates which attributes of a message will be returned by BATS for the remainder of the session. This allows Members to tailor the echoed results to the needs of their system without paying for bandwidth or processing they do not need.

Listing of the return bitfields which are permitted per message is contained in **Return Bitfields Per Message** (§ 6, p. 39).

Field	Offset	Length	Data Type	Description
ParamGroupLength	0	2	Binary	Number of bytes for the parameter group, including this field.
ParamGroupType	2	1	Binary	`0x81`
MessageType	3	1	Binary	Return message type for which the bitfields are being specified (e.g., `0x25` for an Order Acknowledgment V2 message)
NumberOfReturn Bitfields	4	1	Binary	Number of bitfields to follow.
$ReturnBitfield_1$	5	1	Binary	Bitfield identifying fields to return.
⋮				
$ReturnBitfield_n$		1	Binary	Last bitfield.

Example Login Request V2 Message:

Field Name	Hexadecimal	Notes
StartOfMessage	BA BA	Start of message bytes.
MessageLength	43 00	67 bytes
MessageType	37	Login Request V2
MatchingUnit	00	Always 0 for inbound messages
SequenceNumber	00 00 00 00	Always 0 for session level messages
SessionSubID	30 30 30 31	0001
Username	54 45 53 54	TEST
Password	54 45 53 54 49 4E 47 00 00 00	TESTING
NumberOfParam Groups	08	8 parameter groups
ParamGroupLength	14 00	20 bytes for this parameter group
ParamGroupType	80	0x80 = Unit Sequences
NoUnspecified UnitReplay	01	True (replay unspecified units)
NumberOfUnits	03	Three unit/sequence pairs to follow;
$UnitNumber_1$	01	Unit 1
$UnitSequence_1$	4A BB 01 00	Last received sequence of 113,482
$UnitNumber_2$	02	Unit 2
$UnitSequence_2$	00 00 00 00	Last received sequence of 0
$UnitNumber_3$	04	Unit 4
$UnitSequence_3$	79 A1 00 00	Last received sequence of 41,337
ParamGroupLength	08 00	8 bytes for this parameter group
ParamGroupType	81	0x81 = Return Bitfields
MessageType	25	0x25 = Order Acknowledgment V2
NumberOfReturn Bitfields	03	3 bitfields to follow
$ReturnBitfield_1$	00	No bitfields from byte 1
$ReturnBitfield_2$	41	*Symbol*, *Capacity*
$ReturnBitfield_3$	05	*Account*, *ClearingAccount*
ParamGroupLength	0C 00	12 bytes for this parameter group
ParamGroupType	81	0x81 = Return Bitfields
MessageType	2C	0x2C = Order Execution V2
NumberOfReturn Bitfields	07	7 bitfields to follow
$ReturnBitfield_1$	00	No bitfields from byte 1
$ReturnBitfield_2$	41	*Symbol*, *Capacity*
$ReturnBitfield_3$	07	*Account*, *ClearingFirm*, *ClearingAccount*
$ReturnBitfield_4$	00	No bitfields from byte 4
$ReturnBitfield_5$	40	*BaseLiquidityIndicator*
$ReturnBitfield_6$	00	No bitfields from byte 6
$ReturnBitfield_7$	01	*SubLiquidityIndicator*

3.1.2 Logout Request

To end the session, the Member should send a Logout Request message. BATS will finish sending any queued data and finally respond with a Logout message and close the connection.

A Member may simply close the connection without logging out, but may lose any queued messages by doing so.

Logout Request remains unchanged between Versions 1 and 2.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x02`
MatchingUnit	5	1	Binary	Always 0 for inbound (Member to BATS) messages.
SequenceNumber	6	4	Binary	Always 0 for session level messages.

Example Logout Request Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes.
MessageLength	`08 00`	8 bytes
MessageType	`02`	Logout Request
MatchingUnit	`00`	Always 0 for inbound messages
SequenceNumber	`00 00 00 00`	Always 0 for session level messages

3.1.3 Client Heartbeat

See **Heartbeats** (§ 2.4, p. 8) for more information about heartbeats and the session level protocol.

Client Heartbeat remains unchanged between Versions 1 and 2.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x03`
MatchingUnit	5	1	Binary	Always 0 for inbound (Member to BATS) messages.
SequenceNumber	6	4	Binary	Always 0 for session level messages.

Example Client Heartbeat Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes.
MessageLength	`08 00`	8 bytes
MessageType	`03`	Client Heartbeat
MatchingUnit	`00`	Always 0 for inbound messages
SequenceNumber	`00 00 00 00`	Always 0 for session level messages

3.2 BATS to Member

3.2.1 Login Response V2

A Login Response V2 message is sent in response to a Login Request V2 message. On a successful login, the *LoginResponseStatus* will be set to `A`. On a failed login, *LoginResponseStatus* will be set to a value other than `A`, and *LoginResponseText* will be set to an appropriate failure description.

BATS will verify Return Bitfields at login time. If the Return Bitfields in a Return Bitfields Parameter Group are invalid, *LoginResponseStatus* will be set to `F`, and *LoginResponseText* will include a description of which byte and bit are invalid. This is done to ensure that reserved fields are not used, and only options that apply to the local market are set. See **Return Bitfields Per Message** (§ 6, p. 39) for additional information.

Note that two sets of sequence numbers are available on the Login Response V2. The set of sequence numbers in the body are the actual BATS to Member sequence numbers indicating the highest sequence numbers available per matching unit. If specified during login, the Unit Sequences Parameter Group will also be returned which is an echo of the sequence numbers the Member presented during login as the highest received. If these are different, it indicates a gap which will be filled by BATS.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x24`
MatchingUnit	5	1	Binary	Always 0 for session level messages.
SequenceNumber	6	4	Binary	Always 0 for session level messages.
LoginResponse Status	10	1	Alphanumeric	Accepted, or the reason for the rejection. `A` = Login Accepted `N` = Not authorized (invalid username/password) `D` = Session is disabled `B` = Session in use `S` = Invalid session `Q` = Sequence ahead in Login message `I` = Invalid unit given in Login message `F` = Invalid return bitfield in login message `M` = Invalid Login Request message structure
LoginResponse Text	11	60	Text	Human-readable text with additional information about the reason for rejection. For successful logins, this is empty. ASCII NUL (0x00) filled on the right, if necessary.
NoUnspecified UnitReplay	71	1	Binary	Echoed back from the original Login Request V2 message.
LastReceived SequenceNumber	72	4	Binary	Last inbound (Member to BATS) message sequence number processed by BATS.
NumberOfUnits	76	1	Binary	A number, n, of unit/sequence pairs to follow, one per unit. A pair for every unit will be sent, even if no messages have been sent to this port today. For unsuccessful logins, this will be 0.
$UnitNumber_1$		1	Binary	A unit number.
$UnitSequence_1$		4	Binary	Highest available BATS to Member sequence number for the unit.

⋮				
UnitNumber$_n$		1	Binary	A unit number.
UnitSequence$_n$		4	Binary	Highest available BATS to Member sequence number for the unit.
NumberOfParam Groups		1	Binary	Echoed back from the original LOGIN REQUEST V2 message.
ParamGroup$_1$				Echoed back from the original LOGIN REQUEST V2 message.
⋮				
ParamGroup$_n$				Echoed back from the original LOGIN REQUEST V2 message.

Example Login Response V2 Message:

Field Name	**Hexadecimal**	**Notes**
StartOfMessage	`BA BA`	Start of message bytes.
MessageLength	`88 00`	136 bytes
MessageType	`24`	Login Response V2
MatchingUnit	`00`	Always 0 for session messages
SequenceNumber	`00 00 00 00`	Always 0 for session level messages
LoginResponseStatus	`41`	`A` = Login Accepted
LoginResponseText	`41 63 63 65 70 74 65 64 00 00`	`Accepted`
	`00 00 00 00 00 00 00 00 00 00`	(padding)
	`00 00 00 00 00 00 00 00 00 00`	(padding)
	`00 00 00 00 00 00 00 00 00 00`	(padding)
	`00 00 00 00 00 00 00 00 00 00`	(padding)
	`00 00 00 00 00 00 00 00 00 00`	(padding)
NoUnspecified UnitReplay	`00`	False (Replay Unspecified Units)
Last Received Sequence Number	`54 4A 02 00`	Last sequence BATS received of 150,100
NumberOfUnits	`04`	Four unit/sequence pairs to follow.
UnitNumber$_1$	`01`	Unit 1
UnitSequence$_1$	`4A BB 01 00`	Actual last sequence of 113,482
UnitNumber$_2$	`02`	Unit 2
UnitSequence$_2$	`00 00 00 00`	Actual last sequence of 0
UnitNumber$_3$	`03`	Unit 3
UnitSequence$_3$	`00 00 00 00`	Actual last sequence of 0
UnitNumber$_4$	`04`	Unit 4
UnitSequence$_4$	`79 A1 00 00`	Actual last sequence of 41,337
NumberOfParam Groups	`03`	3 parameter groups
ParamGroupLength	`14 00`	20 bytes for this parameter group
ParamGroupType	`80`	`0x80` = Unit Sequences
NoUnspecified UnitReplay	`01`	True (replay unspecified units)
NumberOfUnits	`03`	Three unit/sequence pairs to follow
UnitNumber$_1$	`01`	Unit 1
UnitSequence$_1$	`4A BB 01 00`	Last received sequence of 113,482
UnitNumber$_2$	`02`	Unit 2
UnitSequence$_2$	`00 00 00 00`	Last received sequence of 0
UnitNumber$_3$	`04`	Unit 4
UnitSequence$_3$	`79 A1 00 00`	Last received sequence of 41,337

ParamGroupLength	08 00	8 bytes for this parameter group
ParamGroupType	81	0x81 = Return Bitfields
MessageType	25	0x25 = Order Acknowledgment V2
NumberOfReturn Bitfields	03	3 bitfields to follow
$ReturnBitfield_1$	00	No bitfields from byte 1
$ReturnBitfield_2$	41	*Symbol*, *Capacity*
$ReturnBitfield_3$	05	*Account*, *ClearingAccount*
ParamGroupLength	0C 00	12 bytes for this parameter group
ParamGroupType	81	0x81 = Return Bitfields
MessageType	2C	0x2C = Order Execution V2
NumberOfReturn Bitfields	07	7 bitfields to follow
$ReturnBitfield_1$	00	No bitfields from byte 1
$ReturnBitfield_2$	41	*Symbol*, *Capacity*
$ReturnBitfield_3$	07	*Account*, *ClearingFirm*, *ClearingAccount*
$ReturnBitfield_4$	00	No bitfields from byte 4
$ReturnBitfield_5$	40	*BaseLiquidityIndicator*
$ReturnBitfield_6$	00	No bitfields from byte 6
$ReturnBitfield_7$	01	*SubLiquidityIndicator*

3.2.2 Logout

A LOGOUT is usually sent in response to a LOGOUT REQUEST. Any queued data is transmitted, a LOGOUT is sent, and BATS will close the connection. However, a LOGOUT may also be sent if the Member violates the protocol specification (e.g., by moving backwards in sequence number).

The LOGOUT contains the last transmitted sequence number for each unit, allowing the Member to check that their last received sequence number matches.

LOGOUT remains unchanged between Versions 1 and 2.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be 0xBA 0xBA.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	0x08
MatchingUnit	5	1	Binary	Always 0 for session level messages.
SequenceNumber	6	4	Binary	Always 0 for session level messages.
LogoutReason	10	1	Alphanumeric	The reason why the LOGOUT message was sent. U = User Requested E = End of Day A = Administrative ! = Protocol Violation
LogoutReason Text	11	60	Text	Human-readable text with additional information about the reason for logout. Particularly useful if *LogoutReason* = ! (Protocol Violation).
LastReceived SequenceNumber	71	4	Binary	Last inbound (Member to BATS) message sequence number processed by BATS.

NumberOfUnits	75	1	Binary	A number, n (possibly 0), of unit/sequence pairs to follow, one per unit from which the client has received messages.
$UnitNumber_1$		1	Binary	A unit number.
$UnitSequence_1$		4	Binary	Highest available sequence number for the unit.
⋮				
$UnitNumber_n$		1	Binary	A unit number.
$UnitSequence_n$		4	Binary	Highest available sequence number for the unit.

Example Logout Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes.
MessageLength	`59 00`	89 bytes
MessageType	`08`	Logout
MatchingUnit	`00`	Always 0 for session level messages
SequenceNumber	`00 00 00 00`	Always 0 for session level messages
LogoutReason	`55`	`U` = User Requested
LogoutReason	`55 73 65 72 00 00 00 00 00 00`	`User`
Text	`00 00 00 00 00 00 00 00 00 00` `00 00 00 00 00 00 00 00 00 00` `00 00 00 00 00 00 00 00 00 00` `00 00 00 00 00 00 00 00 00 00` `00 00 00 00 00 00 00 00 00 00`	
LastReceived SequenceNumber	`54 5A 02 00`	Last BATS received sequence of 150,100
NumberOfUnits	`03`	Three unit/sequence pairs to follow.
$UnitNumber_1$	`01`	Unit 1
$UnitSequence_1$	`4A BB 01 00`	Last sent sequence of 113,482
$UnitNumber_2$	`02`	Unit 2
$UnitSequence_2$	`00 00 00 00`	Last sent sequence of 0
$UnitNumber_3$	`04`	Unit 4
$UnitSequence_3$	`79 A1 00 00`	Last sent sequence of 41,337

3.2.3 Server Heartbeat

See **Heartbeats** (§ 2.4, p. 8) for more information about heartbeats and the session level protocol.

Server Heartbeat remains unchanged between Versions 1 and 2.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x09`
MatchingUnit	5	1	Binary	Always 0 for session level messages.
SequenceNumber	6	4	Binary	Always 0 for session level messages.

Example Server Heartbeat Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes.
MessageLength	`08 00`	8 bytes
MessageType	`09`	Server Heartbeat
MatchingUnit	`00`	Always 0 for session level messages
SequenceNumber	`00 00 00 00`	Always 0 for session level messages

3.2.4 Replay Complete

See **Login, Replay and Sequencing** (§ 2.2, p. 7) for more information on Login, sequencing and replay. Replay Complete remains unchanged between Versions 1 and 2.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x13`
MatchingUnit	5	1	Binary	Always 0 for session level messages.
SequenceNumber	6	4	Binary	Always 0 for session level messages.

Example Replay Complete Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes.
MessageLength	`08 00`	8 bytes
MessageType	`13`	Replay Complete
MatchingUnit	`00`	Always 0 for session level messages
SequenceNumber	`00 00 00 00`	Always 0 for session level messages

4 Application Messages

4.1 Member to BATS

4.1.1 New Order V2

A New Order V2 message consists of a number of required fields followed by a number of optional fields. The optional fields used are specified by setting bits in the *NewOrderBitfields*. Fields must be appended at the end of the message, starting with the lowest order enabled bit in the first bitfield first.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x38`
MatchingUnit	5	1	Binary	Always 0 for inbound (Member to BATS) messages.
SequenceNumber	6	4	Binary	The sequence number for this message.
ClOrdID	10	20	Text	Corresponds to *ClOrdID* (11) in BATS FIX. Day-unique ID chosen by the client. Characters in the ASCII range 33–126 are allowed, except for comma, semicolon, and pipe. If the ClOrdID matches a live order, the order will be rejected as duplicate. **Note: BATS only enforces uniqueness of ClOrdID values among currently live orders. However, we *strongly* recommend that you keep your ClOrdID values day-unique.**
Side	30	1	Alphanumeric	Corresponds to *Side* (54) in BATS FIX. `1` = Buy `2` = Sell `5` = Sell Short (client affirms ability to borrow) `6` = Sell Short Exempt
OrderQty	31	4	Binary	Corresponds to *OrderQty* (38) in BATS FIX. Order quantity. System limit is 999,999 shares.
NumberOf NewOrder Bitfields	35	1	Binary	Bitfield identifying which bitfields are set. Field values must be appended to the end of the message.
$NewOrderBitfield_1$	36	1	Binary	Bitfield identifying fields to follow.
⋮				
$NewOrderBitfield_n$		1	Binary	Last bitfield.
Optional fields...				

Example New Order V2 Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes.
MessageLength	`4A 00`	73 bytes
MessageType	`38`	New Order V2
MatchingUnit	`00`	Always 0 for inbound messages
SequenceNumber	`64 00 00 00`	Sequence number 100
ClOrdID	`41 42 43 31 32 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00`	`ABC123`
Side	`31`	Buy
OrderQty	`E8 03 00 00`	1,000 shares
NumberOfNewOrder Bitfields	`03`	3 bitfields to follow
NewOrderBitfield1	`04`	*Price*
NewOrderBitfield2	`C1`	*Symbol*, *Capacity*, *RoutingInst*
NewOrderBitfield3	`01`	*Account*
Price	`44 D6 12 00 00 00 00 00`	`123.4500`
Symbol	`4D 53 46 54 00 00 00 00`	`MSFT`
Capacity	`50`	`P` = Principal
RoutingInst	`52 00 00 00`	`R` = Routable
Account	`44 45 46 47 00 00 00 00 00 00 00 00 00 00 00 00`	`DEFG`

4.1.2 Cancel Order V2

Request to cancel an order using the *ClOrdID* from a previous order.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x39`
MatchingUnit	5	1	Binary	Always 0 for inbound (Member to BATS) messages.
SequenceNumber	6	4	Binary	The sequence number for this message.
OrigClOrdID	10	20	Text	Corresponds to *OrigClOrdID* (41) in BATS FIX. *ClOrdID* of the order to cancel.
NumberOf CancelOrder Bitfields	30	1	Binary	Bitfield identifying bitfields which are set. May be 0. Field values must be appended to the end of the message.
CancelOrder Bitfield$_1$	31	1	Binary	Bitfield identifying fields to follow. Only present if *NumberOfCancelOrderBitfields* is non-zero.
⋮				
CancelOrder Bitfield$_n$		1	Binary	Last bitfield.
Optional fields...				

Example Cancel Order V2 Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes
MessageLength	`22 00`	34 bytes
MessageType	`39`	Cancel Order V2
MatchingUnit	`00`	Always 0 for inbound messages
SequenceNumber	`64 00 00 00`	Sequence Number 100
OrigClOrdID	`41 42 43 31 32 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00`	`ABC123`
NumberOfCancel OrderBitfields	`01`	1 bitfield to follow
CancelOrder Bitfield1	`01`	*ClearingFirm*
ClearingFirm	`54 45 53 54`	`TEST`

4.1.3 Modify Order V2

Request to modify an order. The order attributes to be modified are selected using *NumberOfModifyBitfields* and some number of bitfields to follow.

Only *Price*, *Side*, *OrderQty*, *StopPx*, *MaxFloor*, and *OrdType* may be adjusted. Any change in *Price* or increase in *OrderQty* will result in the order losing its time priority. *OrdType* may be adjusted from Limit to Market (and vice versa) but not from Limit to Peg or Peg to Limit). *Side* may only be used to change an order from a short sell to a long sell or vice versa. Modification of *Side* will only result in a loss of priority if *Side* is changed to/from a short sell and the security is in a Regulation SHO Short Sale Circuit Breaker.

Other fields (including *ExecInst* **will be ignored**, and the value from the original order will be reused. In particular, note that when a Day ISO is modified, the ISO designation is applied to the new order.

A change in *MaxFloor* takes effect on the next reserve reload.

Changes in *OrderQty* result in an adjustment of the current order's *OrderQty*. The new *OrderQty* does not directly replace the current order's *LeavesQty*. Rather, a delta is computed from the current *OrderQty* and the replacement *OrderQty*. This delta is then applied to the current *LeavesQty*. If the resulting *LeavesQty* is less than or equal to zero, the order is cancelled. This results in safer behavior when the modification request overlaps partial fills for the current order, leaving the Member in total control of the share exposure of the order.

DiscretionAmount are preserved from the original order and applied to the new size and price.

A Modify Order V2 should not be issued until the Order Modified message for the previous Modify Order V2 has been received. The BOE handler will reject a new Modify Order V2 if it has not seen the result of the prior modification from the Matching Engine. However, Modify Order V2 requests that merely reduce *OrderQty may* be overlapped if the existing *ClOrdID* is reused, as long as the trading identifer has not been opted-in to daily limit trading risk controls. This is the only case where reuse of the *ClOrdID* is allowed.

***OrderQty* must be present on all Modify Order V2 requests.** Messages sent without *OrderQty* will be rejected. To maintain compatibility with Version 1 Modify Order messages, this field remains in the optional block.

***Price* must be present on all Modify Order V2 requests.** Messages sent without *Price* will be rejected. To maintain compatibility with Version 1 Modify Order messages, this field remains in the optional block.

ClearingFirm is required for service bureau ports.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.

MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x3A`
MatchingUnit	5	1	Binary	Always 0 for inbound (Member to BATS) messages.
SequenceNumber	6	4	Binary	The sequence number for this message.
ClOrdID	10	20	Text	New *ClOrdID* for this order.
OrigClOrdID	30	20	Text	Corresponds to *OrigClOrdID* (41) in BATS FIX. *ClOrdID* of the order to replace. In the case of multiple changes to a single order, this will be the *ClOrdID* of the most recently accepted change.
NumberOf ModifyOrder Bitfields	50	1	Binary	Bitfield identifying bitfields which are set. May be 0. Field values must be appended to the end of the message.
ModifyOrder Bitfield$_1$	51	1	Binary	Bitfield identifying fields to follow.
⋮				
ModifyOrder Bitfield$_n$		1	Binary	Last bitfield.
Optional fields...				

Example Modify Order V2 Message:

Field Name	**Hexadecimal**	**Notes**
StartOfMessage	`BA BA`	Start of message bytes
MessageLength	`3E 00`	62 bytes
MessageType	`3A`	Modify Order V2
MatchingUnit	`00`	Always 0 for inbound messages
SequenceNumber	`64 00 00 00`	Sequence Number 100
ClOrdID	`41 42 43 31 32 34 00 00 00 00 00 00 00 00 00 00 00 00 00 00`	`ABC124`
OrigClOrdID	`41 42 43 31 32 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00`	`ABC123`
NumberOfModify OrderBitfields	`01`	1 bitfield to follow
ModifyOrder Bitfield1	`0C`	*OrderQty*, *Price*
OrderQty	`E0 2E 00 00`	12,000 shares
Price	`08 E2 01 00 00 00 00 00`	12.34

4.2 BATS to Member

4.2.1 Order Acknowledgment V2

Order Acknowledgment V2 messages are sent in response to a New Order V2 message. The message corresponds to a FIX Execution Report with *ExecType* (150) = `0` (New).

Per the instructions given in a Return Bitfields Parameter Group on the Login Request V2 (§ 3.1.1, p. 10), optional fields may be appended to echo back information provided in the original New Order V2 message. Fields which have been requested to be echoed back but which were not filled in will still be sent, but filled with binary zero (0x00).

Permitted return bits are described in § 6.1, p. 39.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x25`
MatchingUnit	5	1	Binary	The matching unit which created this message. Matching units in BOE correspond to matching units on Multicast PITCH.
SequenceNumber	6	4	Binary	The sequence number for this message. Distinct per matching unit.
TransactionTime	10	8	DateTime	The time the event occurred in the BATS matching engine (not the time the message was sent).
ClOrdID	18	20	Text	Echoed back from the original order.
OrderID	38	8	Binary	Corresponds to *OrderID* (37) in BATS FIX. Order identifier supplied by BATS. This identifier corresponds to the identifiers used in BATS market data products.
ReservedInternal	46	1	Binary	Reserved for BATS' internal use.
NumberOfReturn Bitfields	47	1	Binary	Number of bitfields to follow.
$ReturnBitfield_1$	48	1	Binary	Bitfield identifying fields to return.
⋮				
$ReturnBitfield_n$		1	Binary	Last bitfield.
Optional fields...				

Example Order Acknowledgment V2 Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes.
MessageLength	`4D 00`	77 bytes
MessageType	`25`	Order Acknowledgment V2
MatchingUnit	`03`	Matching Unit 3
SequenceNumber	`64 00 00 00`	Sequence number 100
TransactionTime	`E0 FA 20 F7 36 71 F8 11`	1,294,909,373,757,324,000
ClOrdID	`41 42 43 31 32 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00`	`ABC123`
OrderID	`05 10 1E B7 5E 39 2F 02`	`171WC1000005` (base 36)
ReservedInternal	`00`	Ignore
NumberOfReturn	`03`	3 bitfields to follow

Bitfields		
ReturnBitfield$_1$	`00`	No bitfields from byte 1
ReturnBitfield$_2$	`41`	*Symbol*, *Capacity*
ReturnBitfield$_3$	`05`	*Account*, *ClearingAccount*
Symbol	`4D 53 46 54 00 00 00 00`	`MSFT`
Capacity	`50`	`0x50` = P = Principal
Account	`41 42 43 00 00 00 00 00` `00 00 00 00 00 00 00 00`	`ABC`
ClearingAccount	`00 00 00 00`	(empty)

Example Minimal Order Acknowledgment V2 Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes.
MessageLength	`2E 00`	46 bytes
MessageType	`25`	Order Acknowledgment V2
MatchingUnit	`03`	Matching Unit 3
SequenceNumber	`64 00 00 00`	Sequence number 100
TransactionTime	`E0 FA 20 F7 36 71 F8 11`	1,294,909,373,757,324,000
ClOrdID	`41 42 43 31 32 33 00 00 00 00` `00 00 00 00 00 00 00 00 00 00`	`ABC123`
OrderID	`05 10 1E B7 5E 39 2F 02`	`171WC1000005` (base 36)
ReservedInternal	`00`	Ignore
NumberOfReturn Bitfields	`00`	No bitfields to follow

4.2.2 Order Rejected V2

Order Rejected V2 messages are sent in response to a New Order V2 which must be rejected. This message corresponds to a FIX Execution Report with *ExecType* (150) = `8` (Rejected). Order Rejected V2 messages are unsequenced.

Permitted return bits are described in § 6.2, p. 41.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x26`
MatchingUnit	5	1	Binary	Unsequenced application message. Matching unit will be set to 0.
SequenceNumber	6	4	Binary	Unsequenced application message. Sequence number will be set to 0.
TransactionTime	10	8	DateTime	The time the event occurred in the BATS matching engine (not the time the message was sent).
ClOrdID	18	20	Text	Echoed back from the original order.
OrderRejectReason	38	1	Text	Reason for an order rejection. See **Reason Codes** (§ 8, p. 67) for a list of possible reasons.

Text	39	60	Text	Human readable text with more information about the reject reason.
ReservedInternal	99	1	Binary	Reserved for BATS' internal use.
NumberOfReturn Bitfields	100	1	Binary	Number of bitfields to follow.
$ReturnBitfield_1$	101	1	Binary	Bitfield identifying fields to return.
⋮				
$ReturnBitfield_n$		1	Binary	Last bitfield.
Optional fields...				

Example Order Rejected V2 Message:

Field Name	Hexadecimal	Notes
StartOfMessage	BA BA	Start of message bytes
MessageLength	76 00	118 bytes
MessageType	26	Order Rejected V2
MatchingUnit	00	Unsequenced Message, unit = 0
SequenceNumber	00 00 00 00	Unsequenced Message, sequence = 0
TransactionTime	E0 FA 20 F7 36 71 F8 11	1,294,909,373,757,324,000
ClOrdID	41 42 43 31 32 33 00 00 00 00	ABC123
	00 00 00 00 00 00 00 00 00 00	
OrderRejectReason	44	D
Text	44 75 70 6C 69 63 61 74 65 20	Duplicate ClOrdID
	43 6C 4F 72 64 49 44 00 00 00	
	00 00 00 00 00 00 00 00 00 00	
	00 00 00 00 00 00 00 00 00 00	
	00 00 00 00 00 00 00 00 00 00	
	00 00 00 00 00 00 00 00 00 00	
ReservedInternal	00	Ignore
NumberOfReturn Bitfields	03	3 bitfields to follow
$ReturnBitfield_1$	00	No bitfields from byte 1
$ReturnBitfield_2$	01	*Symbol*
$ReturnBitfield_3$	06	*ClearingFirm*, *ClearingAccount*
Symbol	4D 53 46 54 00 00 00 00	MSFT
ClearingFirm	54 45 53 54	TEST
ClearingAccount	00 00 00 00	(empty)

4.2.3 Order Modified V2

Order Modified V2 messages are sent in response to a Modify Request V2 to indicate that the order has been successfully modified.

Note: You must opt-in to receiving *LeavesQty* in Order Modified V2 messages. In some cases, the last message to be received on an order's lifecycle will be an Order Modified V2 message. The way to know the order is no longer live is to inspect *LeavesQty*. An example of this would be modification of an order whilst an execution is being generated, resulting in the order being reduced to zero outstanding quantity. To maintain return structure compatibility with Members with Version 1, this field remains in the optional block.

Permitted return bits are described in § 6.3, p. 43.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x27`
MatchingUnit	5	1	Binary	The matching unit which created this message. Matching units in BOE correspond to matching units on Multicast PITCH.
SequenceNumber	6	4	Binary	The sequence number for this message. Distinct per matching unit.
TransactionTime	10	8	DateTime	The time the event occurred in the BATS matching engine (not the time the message was sent).
ClOrdID	18	20	Text	Client order ID. This is the *ClOrdID* from the Modify Order message.
OrderID	38	8	Binary	Corresponds to *OrderID* (37) in BATS FIX. The unique *OrderID*. Modifications do *not* change the *OrderID*.
ReservedInternal	46	1	Binary	Reserved for BATS' internal use.
NumberOfReturn Bitfields	47	1	Binary	Number of bitfields to follow.
$ReturnBitfield_1$	48	1	Binary	Bitfield identifying fields to return.
⋮				
$ReturnBitfield_n$		1	Binary	Last bitfield.
Optional fields...				

Example Order Modified V2 Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes
MessageLength	`35 00`	63 bytes
MessageType	`27`	Order Modified V2
MatchingUnit	`03`	Matching Unit 3
SequenceNumber	`64 00 00 00`	Sequence number 100
TransactionTime	`E0 FA 20 F7 36 71 F8 11`	1,294,909,373,757,324,000
ClOrdID	`41 42 43 31 32 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00`	`ABC123`
OrderID	`05 10 1E B7 5E 39 2F 02`	`171WC1000005` (base 36)
ReservedInternal	`00`	Ignore
NumberOfReturn Bitfields	`05`	5 bitfields to follow
$ReturnBitfield_1$	`04`	*Price*
$ReturnBitfield_2$	`00`	No fields from byte 2
$ReturnBitfield_3$	`00`	No fields from byte 3
$ReturnBitfield_4$	`00`	No fields from byte 4
$ReturnBitfield_5$	`02`	*LeavesQty*
Price	`08 E2 01 00 00 00 00 00`	12.34
LeavesQty	`00 00 00 00`	0 (order done)

4.2.4 Order Restated V2

Order V2 Restated messages are sent to inform the Member that an order has been asynchronously modified for some reason without an explicit Modify Order V2 request having been sent.

Some example (non-exhaustive) reasons for Order Restated V2 messages being sent:

- A reserve (iceberg) order has been reloaded.
- An order's remaining quantity was decremented because of a prevented wash trade.
- A routed order has returned to rest on the book after matching liquidity on another market.

Members should be prepared to accept and apply Order Restated V2 messages for any reason.

The return bitfields indicate the characteristics of the order which have changed. Optional fields will be present at the end of the message with the new values.

Note: You must opt-in to receiving *LeavesQty* in Order Restated V2 messages. In some cases, the last message to be received on an order's lifecycle will be an Order Restated V2 message. The way to know the order is no longer live is to inspect *LeavesQty*. An example of this would be restatement of an order in some cases due to *PreventMatch* being set to `d`. To maintain return structure compatibility with Members with Version 1, this field remains in the optional block.

Permitted return bits are described in § 6.4, p. 45.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x28`
MatchingUnit	5	1	Binary	The matching unit which created this message. Matching units in BOE correspond to matching units on Multicast PITCH.
SequenceNumber	6	4	Binary	The sequence number for this message. Distinct per matching unit.
TransactionTime	10	8	DateTime	The time the event occurred in the BATS matching engine (not the time the message was sent).
ClOrdID	18	20	Text	The *ClOrdID* is the identifier from the open order.
OrderID	38	8	Binary	Corresponds to *OrderID* (37) in BATS FIX. The unique *OrderID*. For informational purposes only. Restatements do *not* change the *OrderID*.
RestatementReason	46	1	Alphanumeric	The reason for this Order Restated message. `R` = Reroute `X` = Locked in cross `W` = Wash `L` = Reload `Q` = Liquidity Updated `S` = Size reduced due to SWP BATS reserves the right to add new values as necessary without prior notice.
ReservedInternal	47	1	Binary	Reserved for BATS' internal use.

NumberOfReturn Bitfields	48	1	Binary	Number of bitfields to follow.
$ReturnBitfield_1$	49	1	Binary	Bitfield identifying fields to return.
⋮				
$ReturnBitfield_n$		1	Binary	Last bitfield.
Optional fields...				

Example Order Restated V2 message for a reserve (iceberg) reload:

Field Name	**Hexadecimal**	**Notes**
StartOfMessage	`BA BA`	Start of message bytes
MessageLength	`41 00`	65 bytes
MessageType	`28`	Order Restated V2
MatchingUnit	`03`	Matching Unit 3
SequenceNumber	`64 00 00 00`	Sequence number 100
TransactionTime	`E0 FA 20 F7 36 71 F8 11`	1,294,909,373,757,324,000
ClOrdID	`41 42 43 31 32 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00`	`ABC123`
OrderID	`05 10 1E B7 5E 39 2F 02`	`171WC1000005` (base 36)
RestatementReason	`4C`	`L` = Reload
ReservedInternal	`00`	Ignore
NumberOfReturn Bitfields	`06`	6 bitfields to follow
$ReturnBitfield_1$	`00`	No fields from byte 1
$ReturnBitfield_2$	`00`	No fields from byte 2
$ReturnBitfield_3$	`00`	No fields from byte 3
$ReturnBitfield_4$	`00`	No fields from byte 4
$ReturnBitfield_5$	`02`	*LeavesQty*
$ReturnBitfield_6$	`01`	*SecondaryOrderId*
LeavesQty	`64 00 00 00`	100 shares
SecondaryOrderID	`0A 10 1E B7 5E 39 2F 02`	`171WC100000A` (base 36)

4.2.5 User Modify Rejected V2

User Modify Rejected V2 messages are sent in response to a Modify Order V2 for an order which cannot be modified. User Modify Rejected V2 messages are unsequenced.

This message corresponds to a FIX Execution Report with *MsgType* (35) = `9` (Order Cancel Reject) and *CxlRejResponseTo* (434) = `2` (Order Cancel/Replace Request).

Permitted return bits are described in § 6.5, p. 47.

Field	**Offset**	**Length**	**Data Type**	**Description**
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x29`
MatchingUnit	5	1	Binary	Unsequenced application message. Matching unit will be set to 0.
SequenceNumber	6	4	Binary	Unsequenced application message. Sequence number will be set to 0.

TransactionTime	10	8	DateTime	The time the event occurred in the BATS matching engine (not the time the message was sent).
ClOrdID	18	20	Text	The *ClOrdID* of the modify request which was rejected.
ModifyReject Reason	38	1	Text	Reason for a modify rejection. See **Reason Codes** (§ 8, p. 67) for a list of possible reasons.
Text	39	60	Text	Human readable text with more information about the reject reason.
ReservedInternal	99	1	Binary	Reserved for BATS' internal use.
NumberOfReturn Bitfields	100	1	Binary	Number of bitfields to follow.
$ReturnBitfield_1$	101	1	Binary	Bitfield identifying fields to return.
⋮				
$ReturnBitfield_n$		1	Binary	Last bitfield.
Optional fields...				

Example User Modify Rejected V2 Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes
MessageLength	`63 00`	99 bytes
MessageType	`29`	User Modify Rejected V2
MatchingUnit	`00`	Unsequenced Message, unit = 0
SequenceNumber	`00 00 00 00`	Unsequenced Message, sequence = 0
TransactionTime	`E0 FA 20 F7 36 71 F8 11`	1,294,909,373,757,324,000
ClOrdID	`41 42 43 31 32 33 00 00 00 00` `00 00 00 00 00 00 00 00 00 00`	`ABC123`
ModifyRejectReason	`50`	Pending Fill

Text	`50 65 6E 64 69 6E 67 00 00 00`
	`00 00 00 00 00 00 00 00 00 00`
	`00 00 00 00 00 00 00 00 00 00`
	`00 00 00 00 00 00 00 00 00 00`
	`00 00 00 00 00 00 00 00 00 00`
	`00 00 00 00 00 00 00 00 00 00`
		`Pending`

ReservedInternal	`00`	Ignore
NumberOfReturn Bitfields	`00`	No optional fields

4.2.6 Order Cancelled V2

An order has been cancelled.

Permitted return bits are described in § 6.6, p. 49.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.

MessageType	4	1	Binary	`0x2A`
MatchingUnit	5	1	Binary	The matching unit which created this message. Matching units in BOE correspond to matching units on Multicast PITCH.
SequenceNumber	6	4	Binary	The sequence number for this message. Distinct per matching unit.
TransactionTime	10	8	DateTime	The time the event occurred in the BATS matching engine (not the time the message was sent).
ClOrdID	18	20	Text	The order which was cancelled.
CancelReason	38	1	Text	Reason for the order cancellation. See **Reason Codes** (§ 8, p. 67) for a list of possible reasons.
ReservedInternal	39	1	Binary	Reserved for BATS' internal use.
NumberOfReturn Bitfields	40	1	Binary	Number of bitfields to follow.
$ReturnBitfield_1$	41	1	Binary	Bitfield identifying fields to return.
⋮				
$ReturnBitfield_n$		1	Binary	Last bitfield.
Optional fields...				

Example Order Cancelled V2 Message:

Field Name	**Hexadecimal**	**Notes**
StartOfMessage	`BA BA`	Start of message bytes
MessageLength	`48 00`	72 bytes
MessageType	`2A`	Order Cancelled V2
MatchingUnit	`03`	Matching Unit 3
SequenceNumber	`64 00 00 00`	Sequence number 100
TransactionTime	`E0 FA 20 F7 36 71 F8 11`	1,294,909,373,757,324,000
ClOrdID	`41 42 43 31 32 33 00 00 00 00` `00 00 00 00 00 00 00 00 00 00`	`ABC123`
CancelReason	`55`	`U` = User Requested
ReservedInternal	`00`	Ignore
NumberOfReturn Bitfields	`05`	5 bitfields to follow
$ReturnBitfield_1$	`00`	No fields from byte 1
$ReturnBitfield_2$	`00`	No fields from byte 2
$ReturnBitfield_3$	`06`	*ClearingFirm*, *ClearingAccount*
$ReturnBitfield_4$	`00`	No fields from byte 2
$ReturnBitfield_5$	`01`	*OrigClOrdID*
ClearingFirm	`54 45 53 54`	`TEST`
ClearingAccount	`31 32 33 34`	`1234`
OrigClOrdID	`41 42 43 31 32 31 00 00 00 00` `00 00 00 00 00 00 00 00 00 00`	`ABC121`

4.2.7 Cancel Rejected V2

A Cancel Rejected V2 message is sent in response to a Cancel Order V2 message to indicate that the cancellation cannot occur. Cancel Rejected V2 messages are unsequenced.

Permitted return bitfields are described in § 6.7, p. 51.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x2B`
MatchingUnit	5	1	Binary	Unsequenced application message. Matching unit will be set to 0.
SequenceNumber	6	4	Binary	Unsequenced application message. Sequence number will be set to 0.
TransactionTime	10	8	DateTime	The time the event occurred in the BATS matching engine (not the time the message was sent).
ClOrdID	18	20	Text	The order whose cancel was rejected.
CancelReject Reason	38	1	Text	Reason for a cancel rejection. See **Reason Codes** (§ 8, p. 67) for a list of possible reasons.
Text	39	60	Text	Human readable text with more information about the reject reason.
ReservedInternal	99	1	Binary	Reserved for BATS' internal use.
NumberOfReturn Bitfields	100	1	Binary	Number of bitfields to follow.
$ReturnBitfield_1$	101	1	Binary	Bitfield identifying fields to return.
⋮				
$ReturnBitfield_n$		1	Binary	Last bitfield.
Optional fields...				

Example Cancel Rejected V2 Message:

Field Name	Hexadecimal	Notes
StartOfMessage	`BA BA`	Start of message bytes
MessageLength	`63 00`	99 bytes
MessageType	`2B`	Cancel Rejected V2
MatchingUnit	`00`	Unsequenced Message, unit = 0
SequenceNumber	`00 00 00 00`	Unsequenced Message, sequence = 0
TransactionTime	`E0 FA 20 F7 36 71 F8 11`	1,294,909,373,757,324,000
ClOrdID	`41 42 43 31 32 33 00 00 00 00` `00 00 00 00 00 00 00 00 00 00`	`ABC123`
CancelRejectReason	`4A`	`J`

Text	`54 4F 4F 20 4C 41 54 45 00 00`
	`00 00 00 00 00 00 00 00 00 00`
	`00 00 00 00 00 00 00 00 00 00`
	`00 00 00 00 00 00 00 00 00 00`
	`00 00 00 00 00 00 00 00 00 00`
	`00 00 00 00 00 00 00 00 00 00`
		TOO LATE

ReservedInternal	`00`	Ignore
NumberOfReturn Bitfields	`00`	No optional fields

4.2.8 Order Execution V2

An Order Execution V2 is sent for each fill on an order.

Version 2 removes the *AccessFee* field, but adds the optional *FeeCode* field. Rather than returning a monetary value indicating the rebate or charge for an execution, the *FeeCode* is an indication of a fee classification corresponding to an item on the venue's fee schedule.

Permitted return bitfields are described in § 6.8, p. 53.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be `0xBA 0xBA`.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	`0x2C`
MatchingUnit	5	1	Binary	The matching unit which created this message. Matching units in BOE correspond to matching units on Multicast PITCH.
SequenceNumber	6	4	Binary	The sequence number for this message. Distinct per matching unit.
TransactionTime	10	8	DateTime	The time the event occurred in the BATS matching engine (not the time the message was sent).
ClOrdID	18	20	Text	Order receiving the execution.
ExecID	38	8	Binary	Corresponds to *ExecID* (17) in BATS FIX. Execution ID. Unique across all matching units on a given day. Note: ExecIDs will be represented on ODROP, FIXDROP and standard DROP ports as base 36 ASCII. Example conversion: **Decimal** / **Base 36** 28294005440239 / A1234B567 76335905726621 / R248BC23H 728557228187 / 09AP05V2Z
LastShares	46	4	Binary	Corresponds to *LastShares* (32) in BATS FIX. Executed share quantity.
LastPx	50	8	Binary Price	Corresponds to *LastPx* (31) in BATS FIX. Price of this fill.
LeavesQty	58	4	Binary	Corresponds to *LeavesQty* (151) in BATS FIX. Quantity still open for further execution. If zero, the order is complete.

BaseLiquidity Indicator	62	1	Alphanumeric	Indicates whether the trade added or removed liquidity, or was routed to another market. Recommended mapping to FIX *LastLiquidityInd*851 is provided. `A` = Added Liquidity (851 = 1) `R` = Removed Liquidity (851 = 2) `X` = Routed to Another Market (851 = 3) `C` = Auction Trade (BZX Only) (851 = 4)
SubLiquidity Indicator	63	1	Alphanumeric	Additional information about an execution. **BATS may add additional values without notice. Members must gracefully ignore unknown values.** ASCII NUL (0x00) = No Additional Information `E` = Trade Added RPI Liquidity (BYX Only) `H` = Trade added hidden liquidity `I` = Trade added hidden liquidity that was price improved `J` = Execution from first order to join the NBBO `S` = Execution from order that set the NBBO `V` = Visible Liquidity Add Trade that was Price Improved `m` = Midpoint Peg Order Added Liquidity
ContraBroker	64	4	Alphanumeric	Corresponds to *ContraBroker* (375) in BATS FIX. `INET` = Routed to Nasdaq `ARCA` = Routed to NYSE ARCA `AMEX` = Routed to NYSE AMEX `BEX` = Routed to Boston `CHX` = Routed to Chicago `NYSE` = Routed to New York `PSX` = Routed to Philadelphia `DRT` = Routed to DRT Pool (formerly DART) `BATS`[1] = BATS BZX Exchange `BYXX`[1] = BATS BYX Exchange `EDGA`[1] = Routed to Direct Edge `EDGX`[1] = Routed to Direct Edge
ReservedInternal	68	1	Binary	Reserved for BATS' internal use.
NumberOfReturn Bitfields	69	1	Binary	Number of bitfields to follow.
$ReturnBitfield_1$	70	1	Binary	Bitfield identifying fields to return.
⋮				
$ReturnBitfield_n$		1	Binary	Last bitfield.
Optional fields...				

[1] Internally matched if *ContraBroker* matches the identifier of the local trading platform's book.

Example Order Execution V2 Message:

Field Name	Hexadecimal	Notes
StartOfMessage	BA BA	Start of message bytes
MessageLength	4F 00	79 bytes
MessageType	2C	Order Execution V2
MatchingUnit	03	Matching Unit 3
SequenceNumber	64 00 00 00	Sequence number 100
TransactionTime	E0 FA 20 F7 36 71 F8 11	1,294,909,373,757,324,000
ClOrdID	41 42 43 31 32 33 00 00 00 00	ABC123
	00 00 00 00 00 00 00 00 00 00	
ExecID	01 F0 B7 D9 71 21 00 00	D19800001 (base 36)
LastShares	64 00 00 00	100 shares
LastPx	08 E2 01 00 00 00 00 00	12.34
BaseLiquidityIndicator	41	A = Added
SubLiquidityIndicator	00	(unset)
ContraBroker	42 41 54 53	BATS
ReservedInternal	00	Ignore
NumberOfReturn Bitfields	03	3 bitfields to follow
$ReturnBitfield_1$	00	No bitfields from byte 1
$ReturnBitfield_2$	00	No bitfields from byte 2
$ReturnBitfield_3$	46	*ClearingFirm*, *ClearingAccount*, *OrderQty*
ClearingFirm	54 45 53 54	TEST
ClearingAccount	31 32 33 43	1234
OrderQty	78 00 00 00	120 shares

4.2.9 Trade Cancel or Correct V2

Used to relay a trade which has been cancelled (busted) or corrected (price or size change only). The *CorrectedPrice* and optional *CorrectedSize* fields will be set to 0 for cancelled trades and to the new trade price and/or size for corrected trades. Trade Cancel or Correct V2 can be sent for same day as well as previous day trades.

Permitted return bitfields are described in § 6.9, p. 55.

Field	Offset	Length	Data Type	Description
StartOfMessage	0	2	Binary	Must be 0xBA 0xBA.
MessageLength	2	2	Binary	Number of bytes for the message, including this field but not including the two bytes for the *StartOfMessage* field.
MessageType	4	1	Binary	0x2D
MatchingUnit	5	1	Binary	The matching unit which created this message. Matching units in BOE correspond to matching units on Multicast PITCH.
SequenceNumber	6	4	Binary	The sequence number for this message. Distinct per matching unit.
TransactionTime	10	8	DateTime	The time the event occurred in the BATS matching engine (not the time the message was sent).
ClOrdID	18	20	Text	*ClOrdID* of the order whose fill is being cancelled or corrected.

OrderID	38	8	Binary	Corresponds to *OrderID* (37) in BATS FIX. Order whose fill is being cancelled or corrected.
ExecRefID	46	8	Binary	Corresponds to *ExecRefID* (19) in BATS FIX. Refers to the *ExecID* (o)f the fill being cancelled or corrected.
Side	54	1	Alphanumeric	Side of the order.
BaseLiquidity Indicator	55	1	Alphanumeric	Indicates whether the trade added or removed liquidity, or was routed to another market. Recommended mapping to FIX *LastLiquidityInd*851 is provided. `A` = Added Liquidity (851 = 1) `R` = Removed Liquidity (851 = 2) `X` = Routed to Another Market (851 = 3) `C` = Auction Trade (BZX Only) (851 = 4)
ClearingFirm	56	4	Alpha	Echoed back from the original order.
ClearingAccount	60	4	Text	Echoed back from the original order.
LastShares	64	4	Binary	Number of shares of the trade being cancelled.
LastPx	68	8	Binary Price	Price of the trade being cancelled.
CorrectedPrice	76	8	Binary Price	For trade corrections, this is the new trade price. For trade breaks, this is set to 0.
OrigTime	84	8	DateTime	Corresponds to *OrigTime* (42). The date and time of the original trade, in GMT.
ReservedInternal	92	1	Binary	Reserved for BATS' internal use.
NumberOfReturn Bitfields	93	1	Binary	Number of bitfields to follow.
$ReturnBitfield_1$	94	1	Binary	Bitfield identifying fields to return.
⋮				
$ReturnBitfield_n$		1	Binary	Last bitfield.
Optional fields...				

Example Trade Cancel or Correct V2 Message:

Field Name	**Hexadecimal**	**Notes**
StartOfMessage	`BA BA`	Start of message bytes
MessageLength	`66 00`	102 bytes
MessageType	`2D`	Trade Cancel or Correct V2
MatchingUnit	`03`	Matching Unit 3
SequenceNumber	`64 00 00 00`	Sequence number 100
TransactionTime	`E0 FA 20 F7 36 71 F8 11`	1,294,909,373,757,324,000
ClOrdID	`41 42 43 31 32 33 00 00 00 00` `00 00 00 00 00 00 00 00 00 00`	`ABC123`
OrderID	`05 10 1E B7 5E 39 2F 02`	`171WC1000005` (base 36)
ExecRefID	`01 F0 B7 D9 71 21 00 00`	`D19800001` (base 36)
Side	`31`	Buy
BaseLiquidity Indicator	`41`	`A` = Added
ClearingFirm	`54 45 53 54`	`TEST`
ClearingAccount	`00 00 00 00`	(empty)

LastShares	C4 09 00 00	2,500 shares
LastPx	5C 13 04 00 00 00 00 00	26.71
CorrectedPrice	00 00 00 00 00 00 00 00	0 (cancelled)
OrigTime	E0 BA 75 95 15 4C EB 11	1,291,209,373,757,324,000
ReservedInternal	00	Ignore
NumberOfReturn Bitfields	02	2 bitfields to follow
$ReturnBitfield_1$	00	No fields from byte 1
$ReturnBitfield_2$	01	*Symbol*
Symbol	4D 53 46 54 00 00 00 00	MSFT

5 Input Bitfields Per Message

Legend:

• Indicates that the field can be requested for a message
– Indicates that the field cannot be requested for a message

5.1 New Order V2

Byte	Bit	Field	
1	1	*ClearingFirm*	•
	2	*ClearingAccount*	•
	4	*Price*	•
	8	*ExecInst*	•
	16	*OrdType*	•
	32	*TimeInForce*	•
	64	*MinQty*	•
	128	*MaxFloor*	•
2	1	*Symbol*	•
	2	*SymbolSfx*	•
	4	*Currency*	–
	8	*IdSource*	–
	16	*SecurityId*	–
	32	*SecurityExchange*	–
	64	*Capacity*	•
	128	*RoutingInst*	•
3	1	*Account*	•
	2	*DisplayIndicator*	•
	4	*MaxRemovePct*	•
	8	*DiscretionAmount*	•
	16	*PegDifference*	•
	32	*PreventMatch*	•
	64	*LocateRequired*	•
	128	*ExpireTime*	•
4	1	*MaturityDate*	–
	2	*StrikePrice*	–
	4	*PutOrCall*	–
	8	*RiskReset*	–
	16	*OpenClose*	–
	32	*CMTANumber*	–
	64	*TargetPartyID*	–
	128	*Reserved*	–
5	1	*CrossFlag*	–
	2	*AttributedQuote*	•
	4	*BookingType*	–
	8	*ExtExecInst*	•
	16	*Reserved*	–
	32	*Reserved*	–
	64	*Reserved*	–
	128	*Reserved*	–

continued. . .

Byte	Bit	Field	
6	1	*DisplayRange*	•
	2	*StopPx*	•
	4	*RoutStrategy*	•
	8	*RouteDeliveryMethod*	•
	16	*ExDestination*	•
	32	*EchoText*	•
	64	*Reserved*	–
	128	*Reserved*	–

5.2 Cancel Order V2

Byte	Bit	Field	
1	1	*ClearingFirm*	•
	2	*MassCancelLockout*	–
	4	*MassCancel*	–
	8	*OsiRoot*	–
	16	*MassCancelId*	–
	32	*Reserved*	–
	64	*Reserved*	–
	128	*Reserved*	–

ClearingFirm is required for service bureau ports.

5.3 Modify Order V2

Byte	Bit	Field	
1	1	*ClearingFirm*	•
	2	*Reserved*	–
	4	*OrderQty*	⋆
	8	*Price*	⋆
	16	*OrdType*	•
	32	*CancelOrigOnReject*	•
	64	*ExecInst*	•
	128	*Side*	•
2	1	*MaxFloor*	•
	2	*StopPx*	•
	4	*Reserved*	–
	8	*Reserved*	–
	16	*Reserved*	–
	32	*Reserved*	–
	64	*Reserved*	–
	128	*Reserved*	–

⋆ Both *OrderQty* and *Price* must be present on all Modify Order V2 requests. Messages sent without both fields will be rejected. To maintain compatibility with Version 1 Modify Order messages, this field remains in the optional block.

ClearingFirm is required for service bureau ports.

6 Return Bitfields Per Message

Legend:

• Indicates that the field can be requested for a message
– Indicates that the field cannot be requested for a message

6.1 Order Acknowledgment V2

Byte	Bit	Field	
1	1	*Side*	•
	2	*PegDifference*	•
	4	*Price*	•
	8	*ExecInst*	•
	16	*OrdType*	•
	32	*TimeInForce*	•
	64	*MinQty*	•
	128	*MaxRemovePct*	•
2	1	*Symbol*	•
	2	*SymbolSfx*	•
	4	*Currency*	–
	8	*IdSource*	–
	16	*SecurityId*	–
	32	*SecurityExchange*	–
	64	*Capacity*	•
	128	*CrossFlag*	–
3	1	*Account*	•
	2	*ClearingFirm*	•
	4	*ClearingAccount*	•
	8	*DisplayIndicator*	•
	16	*MaxFloor*	•
	32	*DiscretionAmount*	•
	64	*OrderQty*	•
	128	*PreventParticipantMatch*	•
4	1	*MaturityDate*	–
	2	*StrikePrice*	–
	4	*PutOrCall*	–
	8	*OpenClose*	–
	16	*ClOrdIdBatch*	–
	32	*CorrectedSize*	–
	64	*PartyID*	–
	128	*AccessFee*	–
5	1	*OrigClOrdId*	•
	2	*LeavesQty*	•
	4	*LastShares*	•
	8	*LastPrice*	•
	16	*DisplayPrice*	•
	32	*WorkingPrice*	•
	64	*BaseLiquidityIndicator*	•
	128	*ExpireTime*	•

continued. . .

Byte	Bit	Field	
6	1	*SecondaryOrderId*	•
	2	*CCP*	–
	4	*ContraCapacity*	–
	8	*AttributedQuote*	•
	16	*ExtExecInst*	•
	32	*BulkOrderIds*	–
	64	*BulkRejectReasons*	–
	128	*PartyRole*	–
7	1	*SubLiquidityIndicator*	•
	2	*TradeReportTypeReturn*	–
	4	*TradePublishIndReturn*	–
	8	*Text*	–
	16	*Bid*	–
	32	*Offer*	–
	64	*LargeSize*	–
	128	*(Reserved)*	–
8	1	*FeeCode*	–
	2	*EchoText*	•
	4	*StopPx*	•
	8	*RoutingInst*	•
	16	*RoutStrategy*	•
	32	*RouteDeliveryMethod*	•
	64	*ExDestination*	•
	128	*TradeReportRefID*	–
9	1	*MarketingFeeCode*	–
	2	*TargetPartyID*	–
	4	*(Reserved)*	–
	8	*(Reserved)*	–
	16	*(Reserved)*	–
	32	*(Reserved)*	–
	64	*(Reserved)*	–
	128	*(Reserved)*	–

6.2 Order Rejected V2

Byte	Bit	Field	
1	1	*Side*	•
	2	*PegDifference*	•
	4	*Price*	•
	8	*ExecInst*	•
	16	*OrdType*	•
	32	*TimeInForce*	•
	64	*MinQty*	•
	128	*MaxRemovePct*	•
2	1	*Symbol*	•
	2	*SymbolSfx*	•
	4	*Currency*	–
	8	*IdSource*	–
	16	*SecurityId*	–
	32	*SecurityExchange*	–
	64	*Capacity*	•
	128	*CrossFlag*	–
3	1	*Account*	•
	2	*ClearingFirm*	•
	4	*ClearingAccount*	•
	8	*DisplayIndicator*	•
	16	*MaxFloor*	•
	32	*DiscretionAmount*	•
	64	*OrderQty*	•
	128	*PreventParticipantMatch*	•
4	1	*MaturityDate*	–
	2	*StrikePrice*	–
	4	*PutOrCall*	–
	8	*OpenClose*	–
	16	*ClOrdIdBatch*	–
	32	*CorrectedSize*	–
	64	*PartyID*	–
	128	*AccessFee*	–
5	1	*OrigClOrdId*	–
	2	*LeavesQty*	–
	4	*LastShares*	–
	8	*LastPrice*	–
	16	*DisplayPrice*	–
	32	*WorkingPrice*	–
	64	*BaseLiquidityIndicator*	–
	128	*ExpireTime*	–
6	1	*SecondaryOrderId*	•
	2	*CCP*	–
	4	*ContraCapacity*	–
	8	*AttributedQuote*	•
	16	*ExtExecInst*	•
	32	*BulkOrderIds*	–
	64	*BulkRejectReasons*	–
	128	*PartyRole*	–

continued. . .

Byte	Bit	Field	
7	1	*SubLiquidityIndicator*	–
	2	*TradeReportTypeReturn*	–
	4	*TradePublishIndReturn*	–
	8	*Text*	–
	16	*Bid*	–
	32	*Offer*	–
	64	*LargeSize*	–
	128	*(Reserved)*	–
8	1	*FeeCode*	–
	2	*EchoText*	•
	4	*StopPx*	•
	8	*RoutingInst*	•
	16	*RoutStrategy*	•
	32	*RouteDeliveryMethod*	•
	64	*ExDestination*	•
	128	*TradeReportRefID*	–
9	1	*MarketingFeeCode*	–
	2	*TargetPartyID*	–
	4	*(Reserved)*	–
	8	*(Reserved)*	–
	16	*(Reserved)*	–
	32	*(Reserved)*	–
	64	*(Reserved)*	–
	128	*(Reserved)*	–

6.3 Order Modified V2

Byte	Bit	Field	
1	1	*Side*	•
	2	*PegDifference*	•
	4	*Price*	•
	8	*ExecInst*	•
	16	*OrdType*	•
	32	*TimeInForce*	•
	64	*MinQty*	•
	128	*MaxRemovePct*	•
2	1	*Symbol*	–
	2	*SymbolSfx*	–
	4	*Currency*	–
	8	*IdSource*	–
	16	*SecurityId*	–
	32	*SecurityExchange*	–
	64	*Capacity*	–
	128	*CrossFlag*	–
3	1	*Account*	•
	2	*ClearingFirm*	•
	4	*ClearingAccount*	•
	8	*DisplayIndicator*	•
	16	*MaxFloor*	•
	32	*DiscretionAmount*	•
	64	*OrderQty*	•
	128	*PreventParticipantMatch*	•
4	1	*MaturityDate*	–
	2	*StrikePrice*	–
	4	*PutOrCall*	–
	8	*OpenClose*	–
	16	*ClOrdIdBatch*	–
	32	*CorrectedSize*	–
	64	*PartyID*	–
	128	*AccessFee*	–
5	1	*OrigClOrdId*	•
	2	*LeavesQty*	•
	4	*LastShares*	•
	8	*LastPrice*	•
	16	*DisplayPrice*	•
	32	*WorkingPrice*	•
	64	*BaseLiquidityIndicator*	•
	128	*ExpireTime*	•
6	1	*SecondaryOrderId*	•
	2	*CCP*	–
	4	*ContraCapacity*	–
	8	*AttributedQuote*	•
	16	*ExtExecInst*	•
	32	*BulkOrderIds*	–
	64	*BulkRejectReasons*	–
	128	*PartyRole*	–

continued. . .

Byte	Bit	Field	
7	1	*SubLiquidityIndicator*	–
	2	*TradeReportTypeReturn*	–
	4	*TradePublishIndReturn*	–
	8	*Text*	–
	16	*Bid*	–
	32	*Offer*	–
	64	*LargeSize*	–
	128	*(Reserved)*	–
8	1	*FeeCode*	–
	2	*EchoText*	•
	4	*StopPx*	•
	8	*RoutingInst*	•
	16	*RoutStrategy*	•
	32	*RouteDeliveryMethod*	•
	64	*ExDestination*	•
	128	*TradeReportRefID*	–
9	1	*MarketingFeeCode*	–
	2	*TargetPartyID*	–
	4	*(Reserved)*	–
	8	*(Reserved)*	–
	16	*(Reserved)*	–
	32	*(Reserved)*	–
	64	*(Reserved)*	–
	128	*(Reserved)*	–

6.4 Order Restated V2

Byte	Bit	Field	
1	1	*Side*	•
	2	*PegDifference*	•
	4	*Price*	•
	8	*ExecInst*	•
	16	*OrdType*	•
	32	*TimeInForce*	•
	64	*MinQty*	•
	128	*MaxRemovePct*	•
2	1	*Symbol*	•
	2	*SymbolSfx*	•
	4	*Currency*	–
	8	*IdSource*	–
	16	*SecurityId*	–
	32	*SecurityExchange*	–
	64	*Capacity*	•
	128	*CrossFlag*	–
3	1	*Account*	•
	2	*ClearingFirm*	•
	4	*ClearingAccount*	•
	8	*DisplayIndicator*	•
	16	*MaxFloor*	•
	32	*DiscretionAmount*	•
	64	*OrderQty*	•
	128	*PreventParticipantMatch*	•
4	1	*MaturityDate*	–
	2	*StrikePrice*	–
	4	*PutOrCall*	–
	8	*OpenClose*	–
	16	*ClOrdIdBatch*	–
	32	*CorrectedSize*	–
	64	*PartyID*	–
	128	*AccessFee*	–
5	1	*OrigClOrdId*	•
	2	*LeavesQty*	•
	4	*LastShares*	•
	8	*LastPrice*	•
	16	*DisplayPrice*	•
	32	*WorkingPrice*	•
	64	*BaseLiquidityIndicator*	•
	128	*ExpireTime*	•
6	1	*SecondaryOrderId*	•
	2	*CCP*	–
	4	*ContraCapacity*	–
	8	*AttributedQuote*	•
	16	*ExtExecInst*	•
	32	*BulkOrderIds*	–
	64	*BulkRejectReasons*	–
	128	*PartyRole*	–

continued. . .

Byte	Bit	Field	
7	1	*SubLiquidityIndicator*	–
	2	*TradeReportTypeReturn*	–
	4	*TradePublishIndReturn*	–
	8	*Text*	–
	16	*Bid*	–
	32	*Offer*	–
	64	*LargeSize*	–
	128	*(Reserved)*	–
8	1	*FeeCode*	–
	2	*EchoText*	•
	4	*StopPx*	•
	8	*RoutingInst*	•
	16	*RoutStrategy*	•
	32	*RouteDeliveryMethod*	•
	64	*ExDestination*	•
	128	*TradeReportRefID*	–
9	1	*MarketingFeeCode*	–
	2	*TargetPartyID*	–
	4	*(Reserved)*	–
	8	*(Reserved)*	–
	16	*(Reserved)*	–
	32	*(Reserved)*	–
	64	*(Reserved)*	–
	128	*(Reserved)*	–

6.5 User Modify Rejected V2

Byte	Bit	Field	
1	1	*Side*	–
	2	*PegDifference*	–
	4	*Price*	–
	8	*ExecInst*	–
	16	*OrdType*	–
	32	*TimeInForce*	–
	64	*MinQty*	–
	128	*MaxRemovePct*	–
2	1	*Symbol*	–
	2	*SymbolSfx*	–
	4	*Currency*	–
	8	*IdSource*	–
	16	*SecurityId*	–
	32	*SecurityExchange*	–
	64	*Capacity*	–
	128	*CrossFlag*	–
3	1	*Account*	–
	2	*ClearingFirm*	–
	4	*ClearingAccount*	–
	8	*DisplayIndicator*	–
	16	*MaxFloor*	–
	32	*DiscretionAmount*	–
	64	*OrderQty*	–
	128	*PreventParticipantMatch*	–
4	1	*MaturityDate*	–
	2	*StrikePrice*	–
	4	*PutOrCall*	–
	8	*OpenClose*	–
	16	*ClOrdIdBatch*	–
	32	*CorrectedSize*	–
	64	*PartyID*	–
	128	*AccessFee*	–
5	1	*OrigClOrdId*	–
	2	*LeavesQty*	–
	4	*LastShares*	–
	8	*LastPrice*	–
	16	*DisplayPrice*	–
	32	*WorkingPrice*	–
	64	*BaseLiquidityIndicator*	–
	128	*ExpireTime*	–
6	1	*SecondaryOrderId*	–
	2	*CCP*	–
	4	*ContraCapacity*	–
	8	*AttributedQuote*	–
	16	*ExtExecInst*	–
	32	*BulkOrderIds*	–
	64	*BulkRejectReasons*	–
	128	*PartyRole*	–

continued. . .

Byte	Bit	Field	
7	1	*SubLiquidityIndicator*	–
	2	*TradeReportTypeReturn*	–
	4	*TradePublishIndReturn*	–
	8	*Text*	–
	16	*Bid*	–
	32	*Offer*	–
	64	*LargeSize*	–
	128	*(Reserved)*	–
8	1	*FeeCode*	–
	2	*EchoText*	–
	4	*StopPx*	–
	8	*RoutingInst*	–
	16	*RoutStrategy*	–
	32	*RouteDeliveryMethod*	–
	64	*ExDestination*	–
	128	*TradeReportRefID*	–
9	1	*MarketingFeeCode*	–
	2	*TargetPartyID*	–
	4	*(Reserved)*	–
	8	*(Reserved)*	–
	16	*(Reserved)*	–
	32	*(Reserved)*	–
	64	*(Reserved)*	–
	128	*(Reserved)*	–

6.6 Order Cancelled V2

Byte	Bit	Field	
1	1	*Side*	•
	2	*PegDifference*	•
	4	*Price*	•
	8	*ExecInst*	•
	16	*OrdType*	•
	32	*TimeInForce*	•
	64	*MinQty*	•
	128	*MaxRemovePct*	•
2	1	*Symbol*	•
	2	*SymbolSfx*	•
	4	*Currency*	–
	8	*IdSource*	–
	16	*SecurityId*	–
	32	*SecurityExchange*	–
	64	*Capacity*	•
	128	*CrossFlag*	–
3	1	*Account*	•
	2	*ClearingFirm*	•
	4	*ClearingAccount*	•
	8	*DisplayIndicator*	•
	16	*MaxFloor*	•
	32	*DiscretionAmount*	•
	64	*OrderQty*	•
	128	*PreventParticipantMatch*	•
4	1	*MaturityDate*	–
	2	*StrikePrice*	–
	4	*PutOrCall*	–
	8	*OpenClose*	–
	16	*ClOrdIdBatch*	–
	32	*CorrectedSize*	–
	64	*PartyID*	–
	128	*AccessFee*	–
5	1	*OrigClOrdId*	•
	2	*LeavesQty*	•
	4	*LastShares*	•
	8	*LastPrice*	•
	16	*DisplayPrice*	•
	32	*WorkingPrice*	•
	64	*BaseLiquidityIndicator*	•
	128	*ExpireTime*	•
6	1	*SecondaryOrderId*	•
	2	*CCP*	–
	4	*ContraCapacity*	–
	8	*AttributedQuote*	•
	16	*ExtExecInst*	•
	32	*BulkOrderIds*	–
	64	*BulkRejectReasons*	–
	128	*PartyRole*	–

continued. . .

Byte	Bit	Field	
7	1	*SubLiquidityIndicator*	–
	2	*TradeReportTypeReturn*	–
	4	*TradePublishIndReturn*	–
	8	*Text*	–
	16	*Bid*	–
	32	*Offer*	–
	64	*LargeSize*	–
	128	*(Reserved)*	–
8	1	*FeeCode*	–
	2	*EchoText*	•
	4	*StopPx*	•
	8	*RoutingInst*	•
	16	*RoutStrategy*	•
	32	*RouteDeliveryMethod*	•
	64	*ExDestination*	•
	128	*TradeReportRefID*	–
9	1	*MarketingFeeCode*	–
	2	*TargetPartyID*	–
	4	*(Reserved)*	–
	8	*(Reserved)*	–
	16	*(Reserved)*	–
	32	*(Reserved)*	–
	64	*(Reserved)*	–
	128	*(Reserved)*	–

6.7 Cancel Rejected V2

Byte	Bit	Field	
1	1	*Side*	•
	2	*PegDifference*	•
	4	*Price*	•
	8	*ExecInst*	•
	16	*OrdType*	•
	32	*TimeInForce*	•
	64	*MinQty*	•
	128	*MaxRemovePct*	•
2	1	*Symbol*	•
	2	*SymbolSfx*	•
	4	*Currency*	–
	8	*IdSource*	–
	16	*SecurityId*	–
	32	*SecurityExchange*	–
	64	*Capacity*	•
	128	*CrossFlag*	–
3	1	*Account*	–
	2	*ClearingFirm*	–
	4	*ClearingAccount*	–
	8	*DisplayIndicator*	–
	16	*MaxFloor*	–
	32	*DiscretionAmount*	–
	64	*OrderQty*	–
	128	*PreventParticipantMatch*	–
4	1	*MaturityDate*	–
	2	*StrikePrice*	–
	4	*PutOrCall*	–
	8	*OpenClose*	–
	16	*ClOrdIdBatch*	–
	32	*CorrectedSize*	–
	64	*PartyID*	–
	128	*AccessFee*	–
5	1	*OrigClOrdId*	–
	2	*LeavesQty*	–
	4	*LastShares*	–
	8	*LastPrice*	–
	16	*DisplayPrice*	–
	32	*WorkingPrice*	–
	64	*BaseLiquidityIndicator*	–
	128	*ExpireTime*	–
6	1	*SecondaryOrderId*	–
	2	*CCP*	–
	4	*ContraCapacity*	–
	8	*AttributedQuote*	–
	16	*ExtExecInst*	–
	32	*BulkOrderIds*	–
	64	*BulkRejectReasons*	–
	128	*PartyRole*	–

continued...

Byte	Bit	Field	
7	1	*SubLiquidityIndicator*	–
	2	*TradeReportTypeReturn*	–
	4	*TradePublishIndReturn*	–
	8	*Text*	–
	16	*Bid*	–
	32	*Offer*	–
	64	*LargeSize*	–
	128	*(Reserved)*	–
8	1	*FeeCode*	–
	2	*EchoText*	•
	4	*StopPx*	•
	8	*RoutingInst*	–
	16	*RoutStrategy*	–
	32	*RouteDeliveryMethod*	–
	64	*ExDestination*	–
	128	*TradeReportRefID*	–
9	1	*MarketingFeeCode*	–
	2	*TargetPartyID*	–
	4	*(Reserved)*	–
	8	*(Reserved)*	–
	16	*(Reserved)*	–
	32	*(Reserved)*	–
	64	*(Reserved)*	–
	128	*(Reserved)*	–

6.8 Order Execution V2

Byte	Bit	Field	
1	1	*Side*	•
	2	*PegDifference*	•
	4	*Price*	•
	8	*ExecInst*	•
	16	*OrdType*	•
	32	*TimeInForce*	•
	64	*MinQty*	•
	128	*MaxRemovePct*	•
2	1	*Symbol*	•
	2	*SymbolSfx*	•
	4	*Currency*	–
	8	*IdSource*	–
	16	*SecurityId*	–
	32	*SecurityExchange*	–
	64	*Capacity*	•
	128	*CrossFlag*	–
3	1	*Account*	•
	2	*ClearingFirm*	•
	4	*ClearingAccount*	•
	8	*DisplayIndicator*	•
	16	*MaxFloor*	•
	32	*DiscretionAmount*	•
	64	*OrderQty*	•
	128	*PreventParticipantMatch*	•
4	1	*MaturityDate*	–
	2	*StrikePrice*	–
	4	*PutOrCall*	–
	8	*OpenClose*	–
	16	*ClOrdIdBatch*	–
	32	*CorrectedSize*	–
	64	*PartyID*	–
	128	*AccessFee*	–
5	1	*OrigClOrdId*	–
	2	*LeavesQty*	–
	4	*LastShares*	–
	8	*LastPrice*	–
	16	*DisplayPrice*	–
	32	*WorkingPrice*	–
	64	*BaseLiquidityIndicator*	–
	128	*ExpireTime*	–
6	1	*SecondaryOrderId*	–
	2	*CCP*	–
	4	*ContraCapacity*	–
	8	*AttributedQuote*	•
	16	*ExtExecInst*	•
	32	*BulkOrderIds*	–
	64	*BulkRejectReasons*	–
	128	*PartyRole*	–

continued...

Byte	Bit	Field	
7	1	*SubLiquidityIndicator*	–
	2	*TradeReportTypeReturn*	–
	4	*TradePublishIndReturn*	–
	8	*Text*	–
	16	*Bid*	–
	32	*Offer*	–
	64	*LargeSize*	–
	128	*(Reserved)*	–
8	1	*FeeCode*	•
	2	*EchoText*	•
	4	*StopPx*	•
	8	*RoutingInst*	•
	16	*RoutStrategy*	•
	32	*RouteDeliveryMethod*	•
	64	*ExDestination*	•
	128	*TradeReportRefID*	–
9	1	*MarketingFeeCode*	–
	2	*TargetPartyID*	–
	4	*(Reserved)*	–
	8	*(Reserved)*	–
	16	*(Reserved)*	–
	32	*(Reserved)*	–
	64	*(Reserved)*	–
	128	*(Reserved)*	–

6.9 Trade Cancel or Correct V2

Byte	Bit	Field	
1	1	*Side*	–
	2	*PegDifference*	–
	4	*Price*	–
	8	*ExecInst*	–
	16	*OrdType*	–
	32	*TimeInForce*	–
	64	*MinQty*	–
	128	*MaxRemovePct*	–
2	1	*Symbol*	•
	2	*SymbolSfx*	•
	4	*Currency*	–
	8	*IdSource*	–
	16	*SecurityId*	–
	32	*SecurityExchange*	–
	64	*Capacity*	•
	128	*CrossFlag*	–
3	1	*Account*	–
	2	*ClearingFirm*	–
	4	*ClearingAccount*	–
	8	*DisplayIndicator*	–
	16	*MaxFloor*	–
	32	*DiscretionAmount*	–
	64	*OrderQty*	–
	128	*PreventParticipantMatch*	–
4	1	*MaturityDate*	–
	2	*StrikePrice*	–
	4	*PutOrCall*	–
	8	*OpenClose*	–
	16	*ClOrdIdBatch*	–
	32	*CorrectedSize*	–
	64	*PartyID*	–
	128	*AccessFee*	–
5	1	*OrigClOrdId*	–
	2	*LeavesQty*	–
	4	*LastShares*	–
	8	*LastPrice*	–
	16	*DisplayPrice*	–
	32	*WorkingPrice*	–
	64	*BaseLiquidityIndicator*	–
	128	*ExpireTime*	–
6	1	*SecondaryOrderId*	–
	2	*CCP*	–
	4	*ContraCapacity*	–
	8	*AttributedQuote*	–
	16	*ExtExecInst*	–
	32	*BulkOrderIds*	–
	64	*BulkRejectReasons*	–
	128	*PartyRole*	–

continued...

Byte	Bit	Field	
7	1	*SubLiquidityIndicator*	–
	2	*TradeReportTypeReturn*	–
	4	*TradePublishIndReturn*	–
	8	*Text*	–
	16	*Bid*	–
	32	*Offer*	–
	64	*LargeSize*	–
	128	*(Reserved)*	–
8	1	*FeeCode*	–
	2	*EchoText*	–
	4	*StopPx*	–
	8	*RoutingInst*	–
	16	*RoutStrategy*	–
	32	*RouteDeliveryMethod*	–
	64	*ExDestination*	–
	128	*TradeReportRefID*	–
9	1	*MarketingFeeCode*	–
	2	*TargetPartyID*	–
	4	*(Reserved)*	–
	8	*(Reserved)*	–
	16	*(Reserved)*	–
	32	*(Reserved)*	–
	64	*(Reserved)*	–
	128	*(Reserved)*	–

7 List of Optional Fields

The following are descriptions of optional fields which may be sent or received.

Field	Length	Data Type	Description
Account	16	Text	Corresponds to *Account* (1) in BATS FIX. Reflected back on execution reports associated with this order. May be made available in the Member's clearing file. Allowed characters are alphanumeric and colon.
AttributedQuote	1	Alphanumeric	Allows for an order to be attributed to a firm's MPID or optionally `RTAL` (for retail firms) in BATS' market data feeds. The order may also be included in attributed summary information displays related to quote/trade information. Must opt-in to support through the BATS Trade Desk. `N` = Do not attribute firm to this order `Y` = Attribute firm to this order `R` = Attribute `RTAL` to this order
BaseLiquidity Indicator	1	Alphanumeric	Indicates whether the trade added or removed liquidity, or was routed to another market. Recommended mapping to FIX *LastLiquidityInd*851 is provided. `A` = Added Liquidity (851 = 1) `R` = Removed Liquidity (851 = 2) `X` = Routed to Another Market (851 = 3) `C` = Auction Trade (BZX Only) (851 = 4)
CancelOrig OnReject	1	Alpha	Corresponds to *CancelOrigOnReject* (9619) in BATS FIX. Indicates handling of original order on failure to modify. `N` = Leave original order alone. `Y` = Cancel original order if modification fails.
Capacity	1	Alpha	Corresponds to *OrderCapacity* (47) in BATS FIX. `A` = Agency `P` = Principal `R` = Riskless Principal
ClearingAccount	4	Text	Corresponds to *OnBehalfOfSubID* (116) and *ClearingAccount* (440) in BATS FIX. Supplemental identifier. Recorded and made available in execution reports. Available via Drop.
ClearingFirm	4	Alpha	Corresponds to *OnBehalfOfCompID* (115) and *ClearingFirm* (439) in BATS FIX. Firm that will clear trade. Must be allowed NSCC MPID.

DiscretionAmount	2	Binary	Corresponds to *DiscretionAmount* (9622) in BATS FIX. • Two implied decimal places (e.g., `10` = $0.10). • Discretion is implicitly added to bid prices and subtracted from offer prices. • Order will be displayed at *Price*, but can execute in the discretionary range. • A discretionary order will use the minimum amount of discretion necessary to achieve execution. • Maximum range is -9999 to 9999 (i.e., -99.99 to 99.99). May not be used with IOC orders. May not be used with post only orders.
DisplayIndicator	1	Alphanumeric	Corresponds to *DisplayIndicator* (9479) in BATS FIX. **Re-pricing Options:** `V` = Default. As determined by port level setting (defaults to `S`) `P` = Price Adjust `m` = Multiple Price Adjust `R` = Cancel back the order if it cannot be booked and displayed without adjustment `r` = Hidden; cancel back the order if it cannot be booked without adjustment `S` = Display Price Sliding (this is to override an opt-out of Display Price Sliding at the port level) `L` = Display Price Sliding, but reject if order crosses NBBO on entry `M` = Multiple Display Price Sliding **Other Options:** `v` = Visible (for visible peg orders only; others will be rejected) `I` = Invisible (implied Midpoint Peg orders) `N` = No Rescrape At Limit. Applicable only to Fully Routable IOC orders (RoutingInst = `R` and TimeInForce = `3`). After walking the price down to the limit, there will be no final scrape at BATS and the cancel reason code will state `X` (Expired) rather than `N` (No Liquidity).
DisplayPrice	8	Binary Price	Only present when order is fully or partially booked. If the order has to be displayed at a less aggressive price for some reason, then that price will be reported here, otherwise equals price. Present for hidden orders, indicating the price the order would have been displayed at.

DisplayRange	4	Binary	Corresponds to *DisplayRange* (8020) in BATS FIX. Used for random replenishment of reserve orders. Random replenishment establishes a range of possible values for the order quantity that is to be displayed. For example, if *MaxFloor* = 2,000, and *DisplayRange* = 200, the displayed quantity will be selected from one of the following values: 1,800, 1,900, 2,000, 2,100, or 2,200. Must be specified in round lots. New in Version 2.
EchoText	64	Text	Corresponds to *Text* (58) in BATS FIX. Free format text string. May be echoed back on BATS to Member messages.
ExDestination	1	Text	Corresponds to *ExDestination* (100) in BATS FIX. Used to specify the designated away venue for *RoutStrategy* = `DIRC` and for *RoutingInst* = `A` (Post to Away). `A` = NYSE MKT[2] `B` = NASDAQ BX[2] `J` = EDGA[2] `K` = EDGX[3] `M` = CHX `N` = NYSE[2] `P` = NYSE ARCA[2] `Q` = NASDAQ[2] `X` = NASDAQ PSX `Y` = BYX[2] `Z` = BZX[2] New in Version 2.

[2]Post to Away option available for `ROUT`, `ROUX`, and `ROUE` only.
[3]Post to EDGX (for `ROUT`, `ROUD`, `ROUE`, `ROUX`, `ROUZ`, `ROUQ`, `RDOT`, `RDOX`, `ROBB`, `ROCO`, `INET`, `IOCM`, `ICMT`.

ExecInst	1	Text	Corresponds to *ExecInst* (18) in BATS FIX. `f` = Intermarket Sweep (Directed or BATS) `P` = Market Peg (peg Buy to NBBO Offer, peg Sell to NBBO Bid) `Q` = Market Maker Peg (see below) `R` = Primary Peg (peg Buy to NBBO Bid, peg Sell to NBBO Offer) `U` = Supplemental Peg Order `o` = Listing Market Opening (for ROOC strategy only) `c` = Listing Market Close (for ROOC strategy only) `a` = Both Listing Market Open and Close (for ROOC strategy only) `M` = Midpoint (peg to NBBO Midpoint) `m` = Midpoint (peg to NBBO Midpoint, but do not match in event the NBBO locks) `L` = Alternative Midpoint (less aggressive of midpoint and 1 tick inside NBBO) **EDGA:** `d` = Midpoint Discretionary Order **BZX:** `r` = Late (for use with Auction Only orders on BZX only); refer to the BATS US Equities Auction Process specification[4] for more information
ExpireTime	8	DateTime	Corresponds to *ExpireTime* (126) in FIX. Required for *TimeInForce* = `6` orders, specifies the datetime (in UTC) that the order expires.
ExtExecInst	1	Text	Corresponds to *ExtExecInst* (9416) in BATS FIX. `N` = None `R` = Retail Order **BYX Exchange:** `P` = Retail Order (Price Improvement Only) `T` = Retail Price Improving Order
FeeCode	2	Alphanumeric	Indicates fee associated with an execution. Fee codes are published in the pricing schedule. New fee codes may be sent with little to no notice. Members are encouraged to code their systems to accept unknown fee codes.
LastPx	8	Binary Price	Corresponds to *LastPx* (31) in BATS FIX. Price of this fill.
LastShares	4	Binary	Corresponds to *LastShares* (32) in BATS FIX. Executed share quantity.
LeavesQty	4	Binary	Corresponds to *LeavesQty* (151) in BATS FIX. Quantity still open for further execution. If zero, the order is complete.

[4] http://www.batstrading.com/resources/membership/BATS_Auction_Process.pdf

LocateReqd	1	Alpha	Corresponds to *LocateReqd* (114) in BATS FIX. Optional, only processed for Sell Short and Sell Short Exempt orders. `N` = Client affirms ability to borrow (default) `Y` = Client does not affirm ability to borrow (results in a reject)
MaxFloor	4	Binary	Corresponds to *MaxFloor* (111) in BATS FIX. Portion of *OrderQty* to display. The balance is reserve. 0 displays the entire quantity. The displayed quantity of each order at a price level is decremented first. When displayed quantity is fully decremented, it is reloaded up to *MaxFloor* from reserve. Default = `0`
MaxRemovePct	1	Binary	Corresponds to *MaxRemovePct* (9618) in BATS FIX. For Post Only At Limit (RoutingInst = `Q`), what percentage of the order quantity which remains after price improvement may be removed at the limit. Must be 0 for non-Post Only At Limit orders. 0 = Don't remove any shares at limit price. 100 = Remove any amount at limit price. If sent, must be 0 on EDGA and EDGX.
MinQty	4	Binary	Corresponds to *MinQty* (110) in BATS FIX. Minimum fill quantity for Book Only hidden or IOC orders which only interact with liquidity on the target book. Ignored for other orders. On entry and user modification, the behaviour is configurable on the port and can apply to the **total** fill size, which may be made up of several **consecutive** smaller fills.
OrderQty	4	Binary	Corresponds to *OrderQty* (38) in BATS FIX. Order quantity. System limit is 999,999 shares.

OrdType	1	Alphanumeric	Corresponds to *OrdType* (40) in BATS FIX. `1` = Market `2` = Limit (default) `3` = Stop `4` = Stop Limit `P` = Pegged Pegged requires *ExecInst* be set to `L`, `M`, `m`, `P`, `Q`, or `R`. Market implies a *TimeInForce* of Day. Market day orders post in LU/LD straddle state or if a short sale during a Regulation SHO short sale circuit breaker. Stop/Stop Limit orders must have *TimeInForce* = `R` (Regular Hours Only) or `0` (Day). Pegged orders by *not* be routable except for midpoint peg orders on EDGA where *RoutStrategy* = `RMPT`.
OrigClOrdID	20	Text	Corresponds to *OrigClOrdID* (41) in BATS FIX.
PegDifference	8	Signed Binary Price	Corresponds to *PegDifference* (211) in BATS FIX. **Optional signed** value up to four decimal places[5] is **added** to the result of peg calculation. Previously was required to be only a non-aggressive offset. Must be zero for midpoint peg or non-pegged orders. **On BYX Exchange:** If *ExtExecInst* = `T` (Retail Price Improving order): • May be priced in $0.001 increments. • Must be ≥ 0 for Buy orders. • Must be ≤ 0 for Sell orders.

[5] *PegDifference* is rounded (down for buy, up for sell) to fit the tick size.

PreventMatch	3	Alpha	Corresponds to *PreventMatch* (7928) in BATS FIX. Three characters: 1st character - MTP Modifier: `N` = Cancel Newest `O` = Cancel Oldest `B` = Cancel Both `S` = Cancel Smallest `D` = Decrement Larger/Cancel Smaller `d` = Same as `D` above, but only decrement *LeavesQty*. Do not restate *OrderQty*. 2nd character - Unique ID Level: `F` = Prevent Match at Member Level `M` = Prevent Match at MPID Level 3rd character - Trading Group ID (optional): Member specified alphanumeric value 0–9, A–Z, or a–z. The Unique ID level (character 2) of both orders must match to prevent a trade. If specified on both orders, Trading Group ID (character 3) must match to prevent a trade. The MTP Modifier (character 1) of the inbound order will be honored, except that if the inbound order specifies Decrement and the resting order does not, and the resting order is larger, then both orders will be cancelled. This exception is to protect the order entry software for the resting order from receiving an unexpected restatement message. If order entry software is prepared to handle unexpected restatement messages, this exception may be override at the port level by requesting "Allow MTP Decrement Override" functionality. Users of MTP Modifier `D` or `d` *and* users of "Allow MTP Decrement Override" functionality must be prepared to receive an ORDER RESTATED V2 message that decrements *LeavesQty* (and, for method `D`, *OrdQty* as well).
Price	8	Binary Price	Corresponds to *Price* (44) in BATS FIX. Limit price. Four implied decimal places. Required for limit orders (*OrdType* = `2`). If specified on a market order (*OrdType* = `1`), the order will be rejected. This field is also used to specify an optional cap price for pegged orders.

RouteDelivery Method	3	Text	Equivalent to *RouteDeliveryMethod* (9350) in BATS FIX. `RTI` = Route to Improve, equivalent to BATS Parallel-D (default if not specified) `RTF` = Route to Fill, equivalent to BATS Parallel-2D *Route to Improve:* Ability to receive price improvement will take priority over speed of execution. *Route to Fill:* Speed of execution will take priority over potential price improvement. Only applicable to *RoutStrategy* = `ROUT`, `ROUX`, and `ROUE`.
RoutingInst	4	Text	1**st character:** `B` = Book Only (not routable, will remove from local book) `P` = Post Only (not routable) `Q` = Post Only at Limit (BZX and BYX only; removes liquidity that improves upon limit price and up to *MaxRemovePct* of remaining *OrdQty* at limit price) `R` = Routable `S` = Super Aggressive—Cross or Lock (order will be removed from the book and routed to any quote that is locking or crossing the order) `X` = Aggressive—Cross Only (order will be removed from the book and routed to any quote that is crossing the order) `K` = Super Aggressive When Odd Lot[6] (routable order will be automatically assigned Super Aggressive status when it becomes an odd lot) `A` = Post to Away[7] (post remainder to an away venue specified in *ExDestination* for applicable routing strategies) 2**nd character:** `D` = Eligible to route to DRT/CLC `L` = Route to displayed markets only (To be used with *RoutStrategy* = DIRC, TRIM, TRIM+, TRIM2, TRIM2+, TRIM3, TRIM3+, SLIM, SLIM+.)

[6]Anticipated availability date of September 12, 2014.
[7]Anticipated availability date of October 17, 2014.

RoutStrategy	6	Text	Equivalent to *RoutStrategy* (9400) in BATS FIX. **Please note:** CLC = Comprehensive Liquidity Check (EDGA/EDGX only) DRT = Dark Routing Technique (BZX/BYX only) LCPMC = Low Cost Protected Market Centers **All exchanges:** `INET` = Book + IOC/Day NASDAQ `RDOT` = Book + (CLC/DRT) + IOC/Day NYSE `RDOX` = Book + IOC/Day NYSE `ROOC` = Listing Market Open + Book + (CLC/DRT) + Street + Listing Market Close[8] `ROUT` = Book + (CLC/DRT) + Street (**default if not specified**) `ROUX` = Book + Street `ROUZ` = Book + (CLC/DRT) `SWPA` = ISO Sweep of All Protected Markets, equivalent to BATS Parallel-T `SWPB` = ISO Sweep of All Protected Markets[9] `DIRC`[10] = Book + (CLC/DRT) + Directed IOC or Directed ISO if *ExecInst* = `f` **On EDGA/EDGX:** `ROUC` = Book + (CLC/DRT) + LCPMC + All Other Protected Markets + Posts to EDGX `ROUD` = Book + Fast CLCs `ROUE` = Book + Fast CLCs + Street `ROUQ` = Book + Superfast CLCs **On BYX/EDGA:** `RMPT` = Book + Midpoint IOC Select (CLC/DRT/Lit Venues) + Post to Local Book if non-IOC[11] `IOCM` = Book + Midpoint IOC to EDGX `ICMT` = Book + (CLC/DRT) + Midpoint IOC to EDGX **EDGA:** `ROBB` = Book + IOC BYX + IOC NASDAQ BX `ROCO` = Book + IOC BYX + IOC NASDAQ BX + CLC **BYX:** `TRIM` = BYX + BX + EDGA + (DRT) + NYSE + BZX `TRIM2` = BYX + (DRT) + BX + EDGA `SLIM` = BYX + LCPMC + (DRT) + LCPMC + All other protected markets *(cont'd)*

[8] Can be used with *ExecInst* = `a`, `c`, or `o` to specify listing market opening/closing eligibility.

[9] SWPB orders will be cancelled immediately if the order quantity is not enough to clear all protected quotes at or better than the limit price on the order.

[10] Field *ExDestination* must be populated with *RoutStrategy* = `DIRC`.

[11] `RMPT` must be used in conjunction with Midpoint Peg order type.

RoutStrategy (Cont.)	6	Text	**BZX:** `TRIM` = BZX + BYX + BX + EDGA + (DRT) + NYSE `TRIM-` = BYX[12] + BX + EDGA + (DRT) + NYSE `TRIM2` = BZX + BYX + (DRT) + BX + EDGA `TRIM2-` = BYX[12] + (DRT) + BX + EDGA `TRIM3` = BZX + BYX + (DRT) + BX `TRIM3-` = BYX[12] + (DRT) + BX `SLIM` = BZX + BYX + LCPMC + (DRT) + LCPMC + All other protected markets `SLIM+` = BYX[12] + BZX + LCPMC + (DRT) + LCPMC + All other protected markets
SecondaryOrderID	8	Binary	Corresponds to *SecondaryOrderID* (198) in BATS FIX. Denotes an alternative *OrderID* which is present on BATS market data feeds (for example, to hide that a reserve (iceberg) order has reloaded). Or, *OrderID* of the contra side of a prevented match.
Side	1	Alphanumeric	Corresponds to *Side* (54) in BATS FIX. `1` = Buy `2` = Sell `5` = Sell Short (client affirms ability to borrow) `6` = Sell Short Exempt
StopPx	8	Binary Price	Corresponds to *StopPx* (99) in BATS FIX. Stop price. Required if *OrdType* = `3` (Stop) or `4` (Stop Limit). Stop and Stop Limit orders will only be triggered off Last Sale Eligible trades. New in Version 2.
SubLiquidity Indicator	1	Alphanumeric	Additional information about an execution. **BATS may add additional values without notice. Members must gracefully ignore unknown values.** ASCII NUL (0x00) = No Additional Information `E` = Trade Added RPI Liquidity (BYX Only) `H` = Trade added hidden liquidity `I` = Trade added hidden liquidity that was price improved `J` = Execution from first order to join the NBBO `S` = Execution from order that set the NBBO `V` = Visible Liquidity Add Trade that was Price Improved `m` = Midpoint Peg Order Added Liquidity
Symbol	8	Alphanumeric	Corresponds to *Symbol* (55) in BATS FIX. **Entire BATS format symbol** or symbol root if using CQS or CMS format.
SymbolSfx	8	Alphanumeric	Corresponds to *SymbolSfx* (65) in BATS FIX. CQS or CMS suffix. **Do not send *SymbolSfx* is using BATS format or if the symbol does not have a suffix.**

[12] Route to BYX Exchange prior to scraping BZX Exchange book unless price improvement is available.

TimeInForce	1	Alphanumeric	Corresponds to *TimeInForce* (59) in FIX. `0` = Day `1` = GTC (allowed, but treated as Day) `2` = At the Open (BZX Exchange only and BZX listed securities only) `3` = IOC (Portion not filled immediately is cancelled. Market orders are implicitly IOC.) `4` = FOK (an IOC order where the entire size must be filled, else the order will be cancelled back) `5` = GTX (Expires at end of extended day) `6` = GTD (expires at earlier of specified *ExpireTime* or end of extended day) `7` = At the Close `R` = RHO (Regular Hours Only)
WorkingPrice	8	Binary Price	Only present when order is fully or partially booked. If price had to be adjusted to a less aggressive value for some reason, then the adjusted price will be reported here, otherwise equals price.

8 Reason Codes

The following is a list of all reason codes used. These reason codes are used in a variety of contexts (order cancellations, order rejections, modify rejections, etc.). All reasons are not valid in all contexts.

`A` = Admin
`D` = Duplicate Identifier (e.g., *ClOrdID*)
`H` = Halted
`I` = Incorrect Data Center
`J` = Too late to cancel
`K` = Order Rate Threshold Exceeded
`L` = Price Exceeds Cross Range
`M` = Liquidity Available Exceeds Order Size
`N` = Ran Out of Liquidity to Execute Against
`O` = *ClOrdID* Doesn't Match a Known Order
`P` = Can't Modify an Order That is Pending Fill
`Q` = Waiting For First Trade
`R` = Routing Unavailable
`T` = Fill would trade through the NBBO
`U` = User Requested
`V` = Would Wash
`W` = Add Liquidity Only Order Would Remove
`X` = Order Expired
`Y` = Symbol Not Supported
`Z` = Unforeseen Reason
`r` = Reserve Reload
`m` = Market Access Risk Limit Exceeded
`o` = Max Open Orders Count Exceeded
`s` = Risk Management Symbol Level
`u` = Limit Up/Down
`w` = Would Remove on Unslide
`x` = Crossed Market

`y` = Order Received by BATS During Replay

9 List of Message Types

9.1 Member to BATS

Message Name	Level	Type	Sequenced
Login Request V2	Session	0x37	No
Logout Request	Session	0x02	No
Client Heartbeat	Session	0x03	No
New Order V2	Application	0x38	Yes
Cancel Order V2	Application	0x39	Yes
Modify Order V2	Application	0x3A	Yes

9.2 BATS to Member

Message Name	Level	Type	Sequenced
Login Response V2	Session	0x24	No
Logout	Session	0x08	No
Server Heartbeat	Session	0x09	No
Replay Complete	Session	0x13	No
Order Acknowledgment V2	Application	0x25	Yes
Order Rejected V2	Application	0x26	No
Order Modified V2	Application	0x27	Yes
Order Restated V2	Application	0x28	Yes
User Modify Rejected V2	Application	0x29	No
Order Cancelled V2	Application	0x2A	Yes
Cancel Rejected V2	Application	0x2B	No
Order Execution V2	Application	0x2C	Yes
Trade Cancel or Correct V2	Application	0x2D	Yes

10 Port Attributes

The table below lists BOE port attributes that are configurable on the port or firm level. Changes to these attributes can be made by contacting the BATS Trade Desk.

Attribute	Default	Description
Allowed Clearing Executing Firm ID(s)*	All MPIDs	Executing Firm ID(s) allowed for trading on the port.
Default Executing Firm ID	None	Default Executing Firm ID to use if none is sent on a NEW ORDER V2.
Allow ISO*	Yes	Allows or disallows ISO orders.
Allow Directed ISO*	Yes	Allows or disallows ISO orders directed to other market centers.
Default Routing Instruction†		Specifies a default value for routing. Fields can be overriden at the order level. The defaults are *RoutingInst* = R, *RouteDeliveryMethod* = RTI, and *RoutStrategy* = ROUT.
Maximum Order Size*	25,000	Maximum order quantity.
Maximum Order Dollar Value*	Unlimited	Maximum dollar value per order.
Default Price Sliding†		Default price sliding behavior. Specifies the default value for *DisplayIndicator*. The default on BYX, BZX, and EDGA is S (and EDGX effective 7/6/2015). For EDGX, the default is P (until 7/6/2015).
Default Price Sliding (Hidden Order Override)†		Default price sliding behavior. Specifies the default value for *DisplayIndicator*. The default on BYX, BZX, and EDGA is S (and EDGX effective 7/6/2015). For EDGX, the default is H (until 7/6/2015).

Cancel on Disconnect	Option 1	BATS offers three options for cancelling orders as a result of a session disconnect: 1. Cancel continuous book orders only (default). 2. Cancel all open orders (continuous book and on-open, on-close, and late orders)[13]. 3. Do not cancel any open orders.
Send Trade Breaks^	No	Enables sending of TRADE CANCEL OR CORRECT V2 messages.
Default MTP Value*^†	None	Specifies default value for *PreventMatch*.
Allow MTP Decrement Override*^	No	Overrides the exception that requires both the resting and inbound order to be marked as "Decrement".
Allow Sponsored Participant MTP Control*^	No	Allows Sponsored Participant to override port default for match trade prevention by using *PreventMatch* on the order level.
Cancel on Reject†	No	Cancels an order upon a cancel or modify reject.
Cancel on Halt	No	Cancel open orders for a symbol upon a halt.
Opt-out of PITCH Obfuscation	No	Opt-out all orders from PITCH *OrderID* obfuscation for hidden and reserve orders.
Fat Finger Protection*	None	Orders entered through the NBBO by a specified percentage based limit price tolerance will be rejected. Maximum of 20%.
Reject Orders on DROP Port Disconnect*	No	Allows Member/Sponsoring Firms to associate DROP port(s) to order entry port(s). If all associated DROP ports experience disconnection, new orders will be rejected until at least one DROP port session has been reestablished.
Reject Orders on DROP Port Disconnect*	30 seconds	Only applicable if "Reject Orders on DROP Port Disconnect" has been enabled. When the last associated DROP port has disconnected, begin rejecting orders on the associated order entry port(s) if a DROP session has not been reestablished within this timeout. Minimum value allowed is 20 seconds.
Cancel Open Orders on DROP Port Disconnect*	No	Only applicable if "Reject Orders on DROP Port Disconnect" has been enabled. When the last associated DROP port has disconnected, cancel all associated open orders.
Suppress Cancels on Session Close	No	Suppress system generated cancels at the regular market and session close. Also functional during early close sessions.

[13]If the disconnection occurs during the cut-off period for an auction, on-open, on-close, and late orders that are to participate in the auction will not be cancelled.

Notional Cutoff Aggregation Methods*	None	Gross exposure = CBB + CBO + CEB + CEO Net exposure = |(CEO + CBO) − (CEB + CBB)| On a given port, BATS will calculate and track four values: CBB (Cumulative Notional Booked Bid Value): The sum of limit price × size for all booked buy limit orders. CBO (Cumulative Notional Booked Offer Value): The sum of limit price × size for all booked sell limit orders. CEB (Cumulative Notional Executed Bid Value): The sum of size × trade price for all executed buy orders. CEO (Cumulative Notional Executed Sell Value): The sum of size × trade price on all executed sell orders.
Gross Daily Risk Limit Order Notional Cutoff*	None	Results in rejects for **limit** orders when **gross** exposure of limit orders exceeds this value for this port. Maximum whole dollar value of $1 billion/port.
Gross Daily Risk Market Order Notional Cutoff*	None	Results in rejects for **market** orders when **gross** exposure of limit orders exceeds this value for this port. Maximum whole dollar value of $1 billion/port.
Net Daily Risk Limit Order Notional Cutoff*	None	Results in rejects for **limit** orders when **net** exposure of limit orders exceeds this value for this port. Maximum whole dollar value of $1 billion/port.
Net Daily Risk Market Order Notional Cutoff*	None	Results in rejects for **market** orders when **net** exposure of limit orders exceeds this value for this port. Maximum whole dollar value of $1 billion/port.
Default Attributed Quote*†	(see description)	Default value for *AttributedQuote*. May only override on the order level after executing Attribution Addendum to the Exchange User Aggrement. Once the Addendum has been executed, may default to Yes, No, or `RTAL` by contacting the BATS Trade Desk.
Crossed Market Cancel/Reject	No	Reject new orders when the NBBO in the security is crossed. Routable orders will have any remaining quantity cancelled back when the order returns to the book. Order modifications which cause a loss in priority will result in a cancel of the original order if the NBBO is crossed upon receipt of the modify request.

Send Peg Restatements	Option 1	Send restatements for Peg order movements. 1. No Peg restatements (default). 2. Market Maker Peg orders only. 3. All Peg orders except Market Maker Peg orders. 4. All Peg orders.
Default to Retail Order†%^	None	Default *ExtExecInst* = `R` or `P`.

*Sponsored Participants require written approval from Sponsors to update these settings on ports associated with a Sponsor's MPID.

†Port attribute can be overridden on an order-by-order basis.

%Requires agreement for use of this feature.

^Requires certification.

11 Support

Please email questions or comments regarding this specification to `tradedesk@bats.com`.

Revision History

July 6, 2015	Version 2.1.2 Adjustments now that EDGX functionality changes are live.
June 10, 2015	Version 2.1.1 Adjusted wording for *ExecInst* value of `o`. Added Reason Code value of `T`. Corrected message length of example NEW ORDER V2 message.
May 19, 2015	Version 2.1.0 Functionality modifications to EDGX to align with the other BATS equity exchanges: (effective 7/6/2015) EDGX Midpoint Match translated to Midpoint Peg No Lock, EDGX Hide Not Slide translated to Display Price Sliding, and EDGX price sliding default changes to Display Price Sliding.
March 25, 2015	Version 2.0.23 Corrected TRIM *RoutStrategy* descriptions. Added `s` (Risk Management Symbol Level) reason code.
January 29, 2015	Version 2.0.22 Removed references to `ROLF` and LavaFlow.
January 8, 2015	Version 2.0.21 Corrected ORDER EXECUTION V2 return bitfields to note that *SubLiquidityIndicator* is not allowed—it's already available in the message body. Minor correction of *PreventMatch* text (no functional change). On *DisplayIndicator*, noted that `I` is implied on Midpoint Peg orders only.
December 19, 2014	Version 2.0.20 Correction for *DiscretionAmount*. The documentation incorrectly indicated this is a Signed Binary field when it is actually a Binary field.

December 2, 2014	Version 2.0.19 *MaxRemovePct* will now be allowed on EDGA and EDGX, but must always be 0.
November 17, 2014	Version 2.0.18 No functional changes. Clarified that LavaFlow's representation in *ExDestination* is `l` which is a lowercase `L`.
November 13, 2014	Version 2.0.17 No functional changes. Noted that *SubLiquidityIndicator* can be requested on ORDER EXECUTION V2 messages (even though it's present in the message body and is extraneous). Updated descriptions of *BaseLiquidityIndicator*. Noted that *RoutingInst* = `Q` (Post Only at Limit) is only available on BZX and BYX. Noted that *TimeInForce* = `R` (Regular Hours Only) is available in non-BATS listed securities.
October 10, 2014	Version 2.0.16 Clarified ability to reuse *ClOrdId* with MODIFY ORDERS when daily limit trading risk controls are enabled.
September 29, 2014	Version 2.0.15 Corrections: `ROUC` routing strategy will only be supported on EDGA/EDGX. Modified description of `ROLF` strategy to be Book + IOC LavaFlow.
September 11, 2014	Version 2.0.14 Correction: *ExtExecInst* wasn't marked as allowed for US Equities NEW ORDER V2.
September 9, 2014	Version 2.0.13 Removed *AccessFee* from ORDER EXECUTION V2 allowed return bitfields. Removed Options-specific BULK ORDER ACKNOWLEDGMENT V2 message from Section 6.
September 8, 2014	Version 2.0.12 Corrections in allowed return bitfields. Updated Options-specific fields to match latest version of Options specification. Removed *ContraCapacity* from allowed return bitfields. Removed *ContraBroker* from List of Optional fields.
August 27, 2014	Version 2.0.11 Corrected stages of `RMPT` route strategy.
August 26, 2014	Version 2.0.10 Added Reason Code of `w` (Would Remove on Unslide).
August 22, 2014	Version 2.0.9 Removed *ContraCapacity* which is not available in US Equities. Added Super Aggressive When Odd Lot *RoutingInst* value.
August 15, 2014	Version 2.0.8 Removed text which indicated version 2 was not yet available as it is now live.
August 12, 2014	Version 2.0.7 Added *RestatementReason* value of `S` (size reduced due to SWP). The "Default Price Sliding" value incorrectly indicated `H` for EDGX instead of the correct value of `P`. Corrected description of Market Peg.
July 9, 2014	Version 2.0.6 Corrected instances where *ContraCapacity* and *CorrectedSize* may be requested as optional return fields.

July 7, 2014	Version 2.0.5 Removed all return bits from USER MODIFY REJECTED V2 messages. No optional return fields are allowed. Corrected a number of optional return bits. Added *RoutingInst*, *RoutStrategy*, *RouteDeliveryMethod*, and *ExDestination* as optional return bits (byte 8).
July 3, 2014	Version 2.0.4 Added field descriptions for *FeeCode* and *EchoText*.
July 1, 2014	Version 2.0.3 Corrected *ExecInst* to note that Midpoint Discretion Order will only be available on EDGA. Corrected Cancel on Disconnect options.
June 4, 2014	Version 2.0.2 Removed references to CBSX and NSX. Retail attribute value changes from `RETL` to `RTAL`. Corrected length of *NumberOfParamGroups* to be one byte (not two bytes). Fixed naming inconsistency of *AttributedQuote* sometimes being called *AttributedOrder*. Added send peg restatements and retail order default port attributes. Noted that *StopPx* may be modified.
May 1, 2014	Version 2.0.1 Retail attribution value changed from `RTL` to `RETL`.
April 4, 2014	Version 2.0.0 First Version 2 release.