



Full Order Book Market Data

API Document

## Table of Contents

Introduction.....	3
1) Welcome to BO .....	3
Messaging Overview .....	3
1) Message header .....	3
2) Application Messages .....	3
Symbol Enums .....	4
1) Symbol Enum Description. ....	4
2) Current Instruments supported by BO and their corresponding Symbol Enums .....	5
BOClient Logon/Logout to Admin Server (MDAS) .....	5
1. BOClientLogon -- Client Sending.....	5
3) BOClientLogon/Logout -- Server Response .....	6
4) Possible Login/Logout Reject Codes .....	9
BOClientLogin to the Full Order Book Market Data Server (FOBS).....	9
1) BOClientLogon -- Client Sending.....	9
2) BOClientLogon/Logout -- Server Response .....	11
3) Possible Login/Logout Reject Codes .....	13
Successful login to FOBS.....	13
1) BOInstrumentRequest .....	13
2) Possible BOInstrumentRequest Reject Codes.....	15
3) BOInstrument – Server Response to BOInstrumentRequest.....	15
BOMDSubscribe Msg.....	17
1) Order Types not included in the market data .....	17
2) BOMDSubscribe Message .....	18
BOTransaction.....	18
1) Message Types .....	18
2) Order Types .....	19
3) Attributes.....	20
4) New LIMIT order – Client Sending .....	23
5) Cancel Replace LIMIT Order – Client Sending.....	24
6) Cancel LIMIT Order – Client Sending .....	26
7) Execution/Execution Partial.....	29

## Introduction

### 1) Welcome to BO.

This document is intended to familiarize the user with the BO Binary API. The BO API uses a Binary Messaging Protocol. Binary messaging protocols allow for lower bandwidth usage and faster data processing and higher throughput.

## Messaging Overview

### 1) Message header

1. Each message in the API is preceded by a 4 byte header. This header conveys the following information
  - 1.1. Message type. The first byte of the header is always a character indicating the type of message to follow. Example: data[0] = 'T'. The second byte is reserved to accommodate double character message types.
  - 1.2. Immediately following the 2 byte message type is a 2 byte short integer containing the total length of the message including the 4 byte message header. The length is a short integer (2 bytes) and is found in the 3<sup>rd</sup> and 4<sup>th</sup> bytes of the header.

Field Name	Data type	Data length	Buffer offset	Example Value	Notes
Msg1	Char	1	0	'H'	
Msg2	Chart	1	1		
MsgLength	Short	2	2	161	

### 2) Application Messages

2. Application messages contain the data necessary to perform the operations indicated by the message type. Application messages always contain the 4 byte header followed by the actual data message. The type of message which follows the header is always found in the 1<sup>st</sup> byte of the header as described above. The length of the message including the header is always found in the 3<sup>rd</sup> and 4<sup>th</sup> bytes of the header as described above.
3. Excel spreadsheets for each message in the API are included with the documentation. All spreadsheets include:
  - 3.1. Message name
  - 3.2. Field names
  - 3.3. Data type of each field

- 3.4. Buffer offset of each field in the message buffer for sending/receiving.
- 3.5. Total length of the message
- 3.6. Required fields for client and server
- 3.7. The following messages are included in the API
  - 3.7.1. BOClientLogon
  - 3.7.2. BOHeartbeat
  - 3.7.3. BOTestrequest
  - 3.7.4. BOInstrumentRequest
  - 3.7.5. BOInstrument
  - 3.7.6. BOInvalidMsg
  - 3.7.7. BOReject
  - 3.7.8. BOTransaction

## Symbol Enums

### 1) Symbol Enum Description.

Symbol Enums are replacements for the character based instrument name to a short integer. Symbols enums are included in almost all BO messages pertaining to orders and risk management and market data reporting. Symbol Enums are used to replace hashing functions necessary to find either orders, instrument data or risk information normally associated with instruments in most other trading systems. It is essential they be included in the messages which require them. Failure to include them will result in a reject of the message and wrong symbol enums will result in undefined behavior. After the login is complete, the user can send a BOInstrumentRequest message to the FOBS (Full Order Book Market Data Server) and will receive back a BOInstrument message which contains all information for each instrument including the symbol name and the symbol enum for that symbol name. The BOInstrumentRequest and BOInstrument messages will be covered in detail in a subsequent section.

## 2) Current Instruments supported by BO and their corresponding Symbol Enums

Instrument Name	Symbol Enum
BTCUSD	1
USDUSDT	2
FLYUSDT	3
BTCUSDT	4

3)

## BOClient Logon/Logout to Admin Server (MDAS)

### 1. BOClientLogon -- Client Sending

1. The BOClientLogon message must be sent to the MDAS in order to initiate the logon process (please contact your BO Representative for the IP address and port).
2. Please refer to the BOClientLogon excel spreadsheet for exact fields, datatypes and buffer offsets.
3. The MDAS will respond with a BOClientLogon with logon status and if logon was successful the IP Address and Port of the FOBS (Full Order Book Market Data Server).
4. Only one login session is permitted for a unique account ID and UserName.
5. BO requests that if they user is going to close the connection a BOClientLogon message should be sent with the LogonType set to 2 prior to closing the connection in order to allow the FOBS to close the connection gracefully.
6. BOClientLogon Example Message - Client Sending

Field name	Data Type	Data Length	Buffer Offset	Required Field	Required Value	Example Value	Note#
<b>Msg1</b>	char	1	0	X	H	H	Header
<b>Msg2</b>	char	1	1				Header
<b>MsgLength</b>	short	2	2	X	143	143	Header
<b>LogonType</b>	Short	2	4	X		1	Note 1
<b>Account</b>	Int	4	6	X		253336	Note 2
<b>2FA</b>	char[]	6	10	X		1F6A	Note 4
<b>UserName</b>	char[]	6	16	X		BOU1	Note 2

<b>TradingSessionID</b>	Int	4	22	*			Note 2
<b>PrimaryOESIP</b>	char[]	24	26	*			Note 3
<b>SecondaryOESIP</b>	char[]	24	50	*			Note 3
<b>PrimaryMDIP</b>	char[]	24	74				Not used
<b>SecondaryIP</b>	char[]	24	98				Not used
<b>SendingTime</b>	Long	8	122				Note 5
<b>MsgSeqNum</b>	Int	4	130			1500201	
<b>Key</b>	Int	4	134			432451	
<b>LoginStatus</b>	Short	1	138				
<b>RejectReason</b>	Short	2	140				
<b>RiskMaster</b>	Char	1	142				

1. Notes:

Note 1: LogonType is a short enum, values:

Login	1	
Logout	2	

If the value is not one of the values above, a logout message will be sent and the connection closed.

Note 2: Value assigned by BO. If this is a logout, the TradingSessionID must be supplied.

Note 3: IP address and port of the FOBS will be sent in the server response BOClientLogon message.

Note 4: 2FA is disabled for the testing phase

Note 5: Sending times are in nanoseconds from the epoch, January 1, 1970

### 3) BOClientLogon/Logout -- Server Response

- 1.1. Upon Receipt of a client logon, the server will respond with a BOClientLogon message.
- 1.2. The status of the logon will be in the LoginStatus field, a value of 1 indicates the logon was successful, a value of 2 indicates the login failed and the server will then close the connection. The reason for the unsuccessful login will be in the RejectReason field. Possible values for the reject will be detailed in a section below.

- 1.3. The IP address and port of the FOBS (Full Order Book Market Data Server) will be supplied in the message in order to allow the user to make the connection to the FOBS. Login to the FOBS will be covered in a subsequent section.
- 1.4. The server may initiate logout messages in the following circumstances:
- 1.5. Matching Engine switching to secondary server (if it is deemed operationally more efficient in the event of a loss of the primary Matching Engine, the FOBS may switch to the secondary FOBS where the new primary Matching Engine is located).
- 1.6. Malformed messages which cannot be processed.
- 1.7. BO initiated shutdown of the FOBS
- 1.8. In the event the FOBS becomes unavailable, the user should reconnect to the secondary FOBS for which the IP Address and Port is supplied in the message.
- 1.9. In the event the client sends a logout message, the server will not respond with a logout message but will instead proceed to close the connection.

Field name	Data Type	Data Length	Buffer Offset	Required Field	Required Value	Example Value	Notes
<b>Msg1</b>	char	1	0	X	H	H	Header
<b>Msg2</b>	char	1	1				Header
<b>MsgLength</b>	short	2	2	X	143	143	Header
<b>LogonType</b>	Short	2	4			1	Note 1
<b>Account</b>	Int	4	6	*		253336	Note 2
<b>2FA</b>	char[]	6	10	X		1F6A	Note 3
<b>UserName</b>	char[]	6	16	X		BOU1	Note 2
<b>TradingSessionID</b>	Int	4	22	X		505	Note 2
<b>PrimaryOESIP</b>	char[]	24	26	*		192.0.0.1:43005	Note 4
<b>SecondaryOESIP</b>	char[]	24	50	*		113.50.1.1:43005	Note 4
<b>PrimaryMDIP</b>	char[]	24	74				Not used
<b>SecondaryIP</b>	char[]	24	98				Not used
<b>SendingTime</b>	Long	8	122				Note 5
<b>MsgSeqNum</b>	int	4	130	X		1500201	

<b>Key</b>	Int	4	134			432451	
<b>LoginStatus</b>	short	1	138				Not used
<b>RejectReason</b>	short	2	140	*			Note 6
<b>RiskMaster</b>	char	1	142				Note 7
		Total Length	143				

1. Notes:

Note 1: LogonType is a short enum (see Client logon above for possible values)

Note 2: TradingSessionID, UserName and Account number included in logon response but may or may not be included in logout messages.

Note 3: Disabled for testing

Note 4: Not included in logout messages (LogonType = 2)

Note 5: Sending times are in nanoseconds from the epoch, January 1, 1970

Note 6: If the login was rejected, the reject reason will be in the field RejectReason

Note 7: Not currently used, for future expansion





<b>Msg1</b>	char	1	0	X	H	H	Header
<b>Msg2</b>	char	1	1				Header
<b>MsgLength</b>	short	2	2	X	143	143	Header
<b>LogonType</b>	Short	2	4	X		1	Note 1
<b>Account</b>	Int	4	6	X		253336	Note 2
<b>2FA</b>	char[]	6	10	X		1F6A	Note 4
<b>UserName</b>	char[]	6	16	X		BOU1	Note 2
<b>TradingSessionID</b>	Int	4	22	*			Note 2
<b>PrimaryOESIP</b>	char[]	24	26	*			Note 3
<b>SecondaryOESIP</b>	char[]	24	50	*			Note 3
<b>PrimaryMDIP</b>	char[]	24	74				Not used
<b>SecondaryIP</b>	char[]	24	98				Not used
<b>SendingTime</b>	Long	8	122				Note 5
<b>MsgSeqNum</b>	int	4	130			1500201	
<b>Key</b>	Int	4	134			432451	
<b>LoginStatus</b>	short	1	138				
<b>RejectReason</b>	short	2	140				
<b>RiskMaster</b>	char	1	142				

2. Notes:

Note 1: LogonType is a short enum, values:

Login	1	
Logout	2	

Note 2: Value assigned by BO. If this is a logout, the TradingSessionID must be supplied.

Note 3: IP address and port of the FOBS will be sent in the server response BOClientLogon message.

Note 4: 2FA is disabled for the testing phase

Note 5: Sending times are in nanoseconds from the epoch, January 1, 1970

## 2) BOClientLogon/Logout -- Server Response

- 2.1. Upon Receipt of a client logon, the server will respond with a BOClientLogon message.
- 2.2. The status of the logon will be in the LoginStatus field, a value of 1 indicates the logon was successful, a value of 2 indicates the login failed and the server will then close the connection. The reason for the unsuccessful login will be in the RejectReason field. Possible values for the reject will be detailed in a section below.
- 2.3. The IP address and port of the FOBS (Full Order Book Market Data Server) will be supplied in the message in order to allow the user to make the connection to the FOBS. Login to the FOBS will be covered in a subsequent section.
- 2.4. Upon Receipt of a client logon, the server will respond with a BOClientLogon message.
- 2.5. The status of the logon will be in the LoginStatus field, a value of 1 indicates the logon was successful, a value of 2 indicates the login failed and the server will then close the connection. The reason for the unsuccessful login will be in the RejectReason field. Possible values for the reject will be detailed in a section below.
- 2.6. The IP address and port of the FOBS will be supplied in the message in order to allow the user to make the connection to the FOBS. Login to the FOBS will be covered in a subsequent section.
- 2.7. The server may initiate logout messages in the following circumstances:
  - 2.7.1. Matching Engine switching to secondary server (if it is deemed operationally more efficient in the event of a loss of the primary, the FOBS may switch to the secondary FOBS where the new primary Matching Engine is located).
  - 2.7.2. Malformed messages which cannot be processed.
  - 2.7.3. BO initiated shutdown of the FOBS
- 2.8. In the event the FOBS becomes unavailable, the user should reconnect to the secondary FOBS for which the IP Address and Port is supplied in the message.
- 2.9. In the event the client sends a logout message, the server will not respond with a logout message but will instead proceed to close the connection.

Field name	Data Type	Data Length	Buffer Offset	Required Field	Required Value	Example Value	Notes
Msg1	Char	1	0	X	H	H	Header
Msg2	Char	1	1				Header

<b>MsgLength</b>	Short	2	2	X	143	143	Header
<b>LogonType</b>	Short	2	4			1	Note 1
<b>Account</b>	Int	4	6	*		253336	Note 2
<b>2FA</b>	char[]	6	10	X		1F6A	Note 3
<b>UserName</b>	char[]	6	16	X		BOU1	Note 2
<b>TradingSessionID</b>	Int	4	22	X		505	Note 2
<b>PrimaryOESIP</b>	char[]	24	26	*		192.0.0.1:4300 5	Note 4
<b>SecondaryOESIP</b>	char[]	24	50	*		113.50.1.1:430 05	Note 4
<b>PrimaryMDIP</b>	char[]	24	74				Not used
<b>SecondaryIP</b>	char[]	24	98				Not used
<b>SendingTime</b>	Long	8	122				Note 5
<b>MsgSeqNum</b>	Int	4	130	X		1500201	
<b>Key</b>	Int	4	134			432451	
<b>LoginStatus</b>	Short	1	138				Not used
<b>RejectReason</b>	Short	2	140	*			Note 6
<b>RiskMaster</b>	Char	1	142				Note 7
		Total Length	143				

## 2. Notes:

Note 1: LogonType is a short enum (see Client logon above for possible values)

Note 2: TradingSessionID, UserName and Account number included in logon response but may or may not be included in logout messages.

Note 3: Disabled for testing

Note 4: Not included in logout messages (LogonType = 2)

Note 5: Sending times are in nanoseconds from the epoch, January 1, 1970

Note 6: If the login was rejected, the reject reason will be in the field RejectReason, see section below for possible values

Note 7: Not currently used, for future expansion

### 3) Possible Login/Logout Reject Codes

ERROR CODE	Value
USER_NOT_FOUND	2
ACCOUNT_NOT_FOUND	3
INVALID_KEY	4
ACCOUNT_DISABLED	5
RISK_ACCOUNT_NOT_FOUND	7
MES_NOT_AVAILABLE_TRADING_DISABLED	9
OES_NOT_AVAILABLE_TRADING_DISABLED	10
MDS_NOT_AVAILABLE_TRADING_DISABLED	11
MSG_TYPE_INVALID	12
NO_RISK_DATA	44
MSG_SEQ_NUM_INVALID	52
USER_ALREADY_LGGGED_IN	53
MALFORMED_MSG	54
SENDING_TIME_INVALID	21

4)

## Successful login to FOBS

After a successful login to the FOBS the user should request the instrument data.

### 1) BOInstrumentRequest

- 2.1. The BOInstrumentRequest message should be sent to receive the parameters necessary to trade any instrument supported by BO. The FOBS will respond with a BOInstrument message.

Field name	Data Type	Data Length	Buffer Offset	Required Field	Required Value	Example Value	Notes
Msg1	Char	1	0	X	Y	Y	Header
Msg2	Char	1	1				Header
MsgLength	Short	2	2	X	62	62	Header
MessageType	Short	2	4				Not used

<b>RejectReason</b>	Int	4	6	*			Note 7
<b>Account</b>	int	4	8	X		100700	Note 1
<b>RequestType</b>	short	2	12	X		2	Note 2
<b>Key</b>	Int	4	14	X			Note 2
<b>SymbolName</b>	char[]	24	18				Note 3
<b>SymbolType</b>	short	2	42				Note 4
<b>SymbolEnum</b>	short	2	44				Note 5
<b>TradingSessionID</b>	int	4	46			506	
<b>SendingTime</b>	Long	8	50	x			Note 6
<b>MsgSeqNum</b>	Int	4	58	X		1500201	
		Total Length	62				

2.2.

Note 1: TradingSessionID, UserName and Account number supplied by BO to the user.

Note 2: RequestType is an enum with the following values:

Enum Name	Enum Value (short int)	
ALL	1	Request all instruments
SYMBOL_ENUM	2	Individual instrument

Note 3: Character string name representation an individual instrument if known, if not known the name will be provided in the BOInstrument response

Note 4: SymbolType is a short integer enum with the following possible values:

Enum Name: SymbolType	Enum Value	Enum Data Type: short
SPOT	1	
FUTURES	2	

DERIVATIVE		
------------	--	--

Note 5: Symbol Enum is a short integer with the following possible values:

Enum name: SymbolEnum	Enum value
BTCUSD	1
USDUSDT	2
FLYUSDT	3
BTCUSDT	4

Note 6: Sending times are in nanoseconds from the epoch, January 1, 1970

Note 7: If the request was rejected, the reject reason will be in the field RejectReason, see section below for possible values

## 2) Possible BOInstrumentRequest Reject Codes

ERROR CODE	Value
<b>USER_NOT_FOUND</b>	2
<b>ACCOUNT_NOT_FOUND</b>	3
<b>INVALID_KEY</b>	4
<b>MSG_SEQ_NUM_INVALID</b>	52
<b>SENDING_TIME_INVALID</b>	21

3)

### 3) BOInstrument – Server Response to BOInstrumentRequest

In response to the BOInstrumentRequest, the server will send a BOInstrument message for each instrument.

#### 2.3. BOInstrument message

Field name	Data Type	Data Length	Buffer Offset	Required Field	Required Value	Example Value	Notes
<b>Msg1</b>	Char	1	0	X	Y	Y	Header

<b>Msg2</b>	Char	1	1				Header
<b>MsgLength</b>	Short	2	2	X	62	62	Header
<b>MessageType</b>	Short	2	4	*	Y	Y	Note 6
<b>Padding</b>	Short	2	6	*			Not used
<b>ResponseType</b>	Short	2	8	X		2	Note 1
<b>SymbolEnum</b>	short	2	10	X		1	Note 2
<b>SymbolName</b>	Char[]	24	12	X		BTCUSD	Note3
<b>SymbolType</b>	Short	2	36			1	Note 2
<b>PriceIncrement</b>	double	2	38			.5	Note 4
<b>MinSize</b>	double	2	46			.00001	
<b>MaxSize</b>	double	4	54			1000	
<b>SendingTime</b>	Long	8	62	x			Note 5
<b>MsgSeqNum</b>	Int	4	70	X		1500201	
		Total Length	74				

#### 2.4.

Note 1: ResponseType is a short enum with the following possible values. If the ResponseType is set to SYMBOL\_ENUM, the SymbolEnum field must be set to the value of the symbol enum being requested.

ResponseType Enum	Enum Value (short int)	
SNAPSHOT	1	
SNAPSHOT_START	2	Multi message responses
SNAPSHOT_CONTINUATION	3	Continuation of response
SNAPSHOT_FINISH	4	Last message of the response
UPDATE	5	
UNSOLICITED	6	
POSS_DUP	7	



Note 2: Please see preceding section for possible values for the SymbolEnum and SymbolType

Note 3: Character representation of the instrument name

Note 4: PriceIncrement is the minimum price movement of the instrument.

Note 5: Sending times are in nanoseconds from the epoch, January 1, 1970

Note 6: MessageType is not really required since a message of this type is defined in the first byte of the header.

## BOMDSubscribe Msg

BOMDSubscribe message is used to subscribe to instruments for which market data is provided. The subscribe message will result in a snapshot of the symbol requested. Upon completion of the subscribe message normal updates will be sent to the user. It is the responsibility of the user to maintain the order book upon receiving the snapshot data and the normal messages. Normal updates will be provided as a BOTransaction message.

### 1) Order Types not included in the market data

1. Hidden order types
2. ICE order types will be provided as limit orders with the order quantity set to the size increment
3. Peg and Peg Hidden order types
4. Stop Market and Stop limit order types
5. Trailing stop market and limit order types
6. Sniper market and limit order types
7. OCO orders will be sent as regular LMT order types

## 2) BOMDSubscribe Message

Field name	Data Type	Data Length	Buffer Offset	Required Field	Required Value	Example Value	Notes
Msg1	Char	1	0	X	s	s	Header
Msg2	Char	1	1				Header
MsgLength	Short	2	2	X	34	34	Header
MessageType	Short	2	4	*		MD_SUBSCRIBE	Note 1
SymbolEnum	Short	2	6				Note 2
SubscribeType	Short	2	8				Note 3
Account	Int	4	10	X		100700	
Key	Int	4	14	X		468321	
TradingSessionID	Int	4	18	X		200	
SendTime	UInt64_t	8	22	X		184232029271	Note 3
SeqNum	Int	4	30	X		52488131	

3)

## BOTransaction

BOTransaction message is used to enter new orders, cancel orders and cancel replace orders. The MES (Matching Engine Server) responds to these orders and conveys the results of new orders, cancellations, cancel replaces, partial executions and executions with the BOTransaction message. The message type in the BOTransaction should not be confused with the message type in the message header. The header message type will always be 'T' in byte 1 of the header. The sub message types in the table below will always be in byte 5 of the data buffer. Please refer to the document titled BO Order Entry API Document for a complete description of BOTransaction messages

### 1) Message Types

2.5.

Enum name: MessageType	Enum Value	Notes
ORDER_NEW	1	
CANCEL_REPLACE	2	
MARGIN_CANCEL_REPLACE	3	Note 1

MARGIN_EXECUTE	4	Note 2
ORDER_STATUS	5	
ORDER_CANCEL	6	
MARGIN_CANCEL	7	Note 1
EXECUTION	8	
EXECUTION_PARTIAL	9	
MARGIN_EXECUTION	10	Note 2
MARGIN_PARTIAL_EXECUTION	11	
REJECT	12	Note 3
ORDER_REJECT	13	Note 3
ORDER_ACK	14	
CANCELLED	15	
REPLACED	16	
QUOTE_FILL	17	Note 4
QUOTE_FILL_PARTIAL	18	
MARGIN_REPLACED	19	Note 1
CANCEL_REPLACE_REJECT	20	

2.6.

Note 1: In some instances, the risk engine may ask the matching engine to cancel replace user orders to reduce the desired size to prevent exceeding the available equity

Note 2: In the event the user exceeds their available equity the risk engine will reduce their position until the position is within the available equity

Note 3: Orders may be rejected by the risk engine or the MES due to equity limits or TIF parameters or missing or wrong data in a message field.

Note 4: QUOTE\_FILL and QUOTE\_FILL\_PARTIAL are executions of user orders which have not been placed on the book (orders placed on the book are commonly called resting orders) and cross the top of book price and interact with a resting order.

2) Order Types

Please refer to document .....for a complete description of each order type.

Enum name: OrdType	Enum Value	Notes
--------------------	------------	-------

LMT	1	Please refer to .....
MKT	2	
STOP_MKT	3	
STOP_LMT	4	Not currently implemented
PEG	5	
HIDDEN	6	
PEG_HIDDEN	7	
OCO	8	
ICE	9	
OCO_ICE	10	Not currently implemented
BRACKET	11	Not currently implemented
SNIPER_MKT	12	
SNIPER_LIMIT	13	
TSM	14	
TSL	15	
TPSL_MARKET	16	Not currently implemented
TPSL_LIMIT	17	Not currently implemented

### 3) Attributes

Attributes can be assigned to various orders. Attributes are used to add additional functionality such as display/refresh attributes to affect the volume an order displays at various stages of its execution cycle. The attribute field is a character array consisting of 12 slots which can be used to set one of 12 different attributes. A value of 'Y' in the attribute array indicates this attribute is to be applied to the order in question. A value of 'N' indicates this attribute is to be ignored for the order in question.

#### 2.7. Attributes

Attributes	Value	
POPPED_TYPE	'Y'/'N'	Internal use only
HIDDEN_ATTRIBUTE	'Y'/'N'	Note 1
DISPLAYSIZE_ATTRIBUTE	'Y'/'N'	Note 2
STOPMKT_ATTRIBUTE	'Y'/'N'	Not currently in use

STOPLMT_ATTRIBUTE	'Y'/'N'	Not currently in use
TSL_ATTRIBUTE	'Y'/'N'	Not currently in use
TSM_ATTRIBUTE	'Y'/'N'	Not currently in use
PEG_ATTRIBUTE	'Y'/'N'	Not currently in use
TPSL_ATTRIBUTE	'Y'/'N'	Not currently in use
STATIC_ATTRIBUTE	'Y'/'N'	Not currently in use
VALIDATED	'Y'/'N'	Not currently in use

2.8.

Note 1: The HIDDEN\_ATTRIBUTE makes the order hidden on the book. It is not visible.

Note 2: DISPLAYSIZE\_ATTRIBUTE instructs the Matching Engine to use the field

DisplaySize as the initial volume to display on the book. When this size has been exhausted, the order is popped off the queue and pushed to the back of the price level queue and the size in the RefreshSize field of BOTransaction is used until the total volume of the order is exhausted. After each depletion of the refresh size, the order is once again popped off the price level queue and pushed to the back of the queue until such time as the total order quantity in the BOOrderQty field is zero or the order is cancelled or cancel replaced.

Field name	Data Type	Data Length	Buffer Offset	Require d Field	Required Value	Example Value	Notes
<b>Msg1</b>	Char	1	0	X	T	T	Header
<b>Msg2</b>	Char	1	1				Header
<b>MsgLength</b>	Short	2	2	X	238	238	Header
<b>MessageType</b>	Short	2	4	*		ORDER_NEW	Note 1
<b>Padding</b>	Short	2	6				Not used
<b>Account</b>	Int	4	8	X		100700	
<b>OrderID</b>	Long	8	12	X		46832151	
<b>SymbolEnum</b>	Short	2	20	X		1	Note 2
<b>OrderType</b>	Short	2	22	X		LMT	Note 3
<b>SymbolType</b>	Short	2	24	X		SPOT	
<b>BOPrice</b>	Double	8	26	X		50100.5	Note 4
<b>BOSide</b>	Short	2	34	X		BUY	Note 5

<b>BOOrderQty</b>	Double	8	36		X		2.0	
<b>TIF</b>	Short	2	44		X		GTC	Note 6
<b>StopLimitPrice</b>	Double	8	46					
<b>BOSymbol</b>	char[]	12	54		X		BTCUSD	
<b>OrigOrderID</b>	Long	8	66					
<b>BOCancelShares</b>	Double	8	74		*			
<b>ExecID</b>	Long	8	82		*			
<b>ExecShares</b>	Double	8	90		*			
<b>RemaingQuantity</b>	Double	8	98					
<b>ExecFee</b>	Double	8	106					
<b>ExpirationDate</b>	char[]	12	114					
<b>TraderID</b>	char[]	6	126					Not used
<b>RejectReason</b>	Short	2	132					
<b>SendingTime</b>	uint64_t	8	134		X		1000	
<b>TradingSessionID</b>	Int	4	142		X		506	
<b>Key</b>	Int	4	146		X		42341	Note 8
<b>DisplaySize</b>	Double	8	150		*			
<b>RefreshSize</b>	Double	8	158		*			
<b>Layers</b>	Short	2	166					
<b>SizeIncrement</b>	Double	8	168					
<b>PriceIncrement</b>	Double	8	176					
<b>PriceOffset</b>	Double	8	184					
<b>BOOrigPrice</b>	Double	8	192					
<b>ExecPrice</b>	Double	8	200					
<b>MsgSeqNum</b>	Long	8	208		X		7948888	
<b>TakeProfitPrice</b>	Double	8	216					
<b>TriggerType</b>	Short	2	224					

<b>Attributes</b>	char[]	12	226		*			Note 7

#### 4) New LIMIT order – Client Sending

Note 1: Message types must be valid according to the values listed in the Message Type table above. Since in this example we would like to place a new order on the book, the MsgType field must be set to ORDER\_NEW.

Note 2: Please see previous sections for valid values for the symbol enum.

Note 3: Order types must be a valid value as defined in the OrdType table above. Since in this example we would like to place a limit order on the book, OrdType field should be set to LMT.

Note 4: Price values must be in a price increment according to the value the user received in the BOInstrument message - PriceIncrement field. Example, BTCUSD, symbol enum 1, has a price increment of 0.5. If the user sends a price of 51000.40, this price is invalid since the cents portion of the price is not in a 0.5 increment. The correct price should have been 51000.50 or 51000.00 or 51001.00, all of these are valid values.

Note 5: The valid side fields are:

Enum Name: SIDE	Enum Value	
BUY	1	
SELL	2	

Note 6: TIF valid values:

Enum Name: TIF	Enum Value	
FOK	1	
GTC	2	
IOC	3	
POO	4	
RED	5	
DAY	6	

Note 7: Attributes allow an order to exhibit additional behavior. Currently only the HIDDEN\_ATTRIBUTE and the DISPLAYSIZE\_ATTRIBUTE are available but the other attributes will be available soon. Please see the section above on attribute behavior. In order to set an attribute a user should do something like this:

```
// function accepts the position in the array to set the value and the actual value
BOTransaction.setAttributes(DISPLAYSIZE, 'Y');
```

If the DISPLAYSIZE attribute is set, the DisplaySize and RefreshSize in the message must also be set, if either is not set to a valid value the message will be rejected. If the user sets the HIDDEN\_ATTRIBUTE to 'Y', this order will be hidden.

Note 8: Currently disabled for testing

### 5) Cancel Replace LIMIT Order – Client Sending

User wishes to cancel replace the order sent in the previous section

Field name	Data Type	Data Length	Buffer Offset	Required Field	Required Value	Example Value	Notes
Msg1	Char	1	0	X	T	T	Header
Msg2	Char	1	1				Header
MsgLength	Short	2	2	X	238	238	Header
MessageType	Short	2	4	*		CANCEL_REPLACE	Note 1
Padding	Short	2	6				Not used
Account	Int	4	8	X		100700	
OrderID	Long	8	12	X		46832152	
SymbolEnum	Short	2	20	X		1	Note 2
OrderType	Short	2	22	X		LMT	Note 3
SymbolType	Short	2	24	X		SPOT	
BOPrice	Double	8	26	X		51102.5	Note 4
BOSide	Short	2	34	X		BUY	Note 5
BOOrderQty	Double	8	36	X		3.0	
TIF	Short	2	44	X		GTC	Note 6
StopLimitPrice	Double	8	46				
BOSymbol	char[]	12	54	X		BTCUSD	
OrigOrderID	Long	8	66			46832151	Note 9
BOCancelShares	Double	8	74	*			
ExecID	Long	8	82	*			
ExecShares	Double	8	90	*			
RemainingQuantity	Double	8	98				



<b>ExecFee</b>	Double	8	106					
<b>ExpirationDate</b>	char[]	12	114					
<b>TraderID</b>	char[]	6	126					Not used
<b>RejectReason</b>	Short	2	132					
<b>SendingTime</b>	uint64_t	8	134		X			
<b>TradingSessionID</b>	Int	4	142		X		506	
<b>Key</b>	Int	4	146		X		42341	Note 8
<b>DisplaySize</b>	Double	8	150		*			
<b>RefreshSize</b>	Double	8	158		*			
<b>Layers</b>	Short	2	166					
<b>SizeIncrement</b>	Double	8	168					
<b>PriceIncrement</b>	Double	8	176					
<b>PriceOffset</b>	Double	8	184					
<b>BOOrigPrice</b>	Double	8	192				50100.5	Note 10
<b>ExecPrice</b>	Double	8	200					
<b>MsgSeqNum</b>	Long	8	208		x		7948888	
<b>TakeProfitPrice</b>	Double	8	216					
<b>TriggerType</b>	Short	2	224					
<b>Attributes</b>	char[]	12	226		*			Note 7

Note 1: Message types must be valid according to the values listed in the Message Type table above. Since in this example we would like to place a new order on the book, the MsgType field must be set to ORDER\_NEW.

Note 2: Please see previous sections for valid values for the symbol enum.

Note 3: Order types must be a valid value as defined in the OrdType table above. Since in this example we would like to place a limit order on the book, OrdType field should be set to LMT.

Note 4: Price values must be in a price increment according to the value the user received in the BOInstrument message - PriceIncrement field. Example, BTCUSD, symbol enum 1, has a price increment of 0.5. If the user sends a price of 51000.40, this price is invalid since the cents portion of the price is not in a 0.5 increment. The correct price should have been 51000.50 or 51000.00 or 51001.00, all of these are valid values.

Note 5: The valid side fields are:

Enum Name: SIDE	Enum Value	
BUY	1	
SELL	2	

Note 6: TIF valid values:

Enum Name: TIF	Enum Value	
FOK	1	
GTC	2	
IOC	3	
POO	4	
RED	5	
DAY	6	

Note 7: Attributes allow an order to exhibit additional behavior. Currently only the HIDDEN\_ATTRIBUTE and the DISPLAYSIZE\_ATTRIBUTE are available but the other attributes will be available soon. Please see the section above on attribute behavior. In order to set an attribute a user should do something like this:

```
// function accepts the position in the array to set the value and the actual value
BOTransaction.setAttributes(DISPLAYSIZE, 'Y');
```

If the DISPLAYSIZE attribute is set, the DisplaySize and RefreshSize in the message must also be set, if either is not set to a valid value the message will be rejected. If the user sets the HIDDEN\_ATTRIBUTE to 'Y', this order will be hidden.

Note 8: Currently disabled for testing

## 6) Cancel LIMIT Order – Client Sending

User wishes to cancel replace the order sent in the previous section

Field name	Data Type	Data Length	Buffer Offset	Required Field	Required Value	Example Value	Notes
Msg1	Char	1	0	X	T	T	Header

<b>Msg2</b>	Char	1	1					Header
<b>MsgLength</b>	Short	2	2		X	238	238	Header
<b>MessageType</b>	Short	2	4		*		ORDER_CANCEL	Note 1
<b>Padding</b>	Short	2	6					Not used
<b>Account</b>	Int	4	8		X		100700	
<b>OrderID</b>	Long	8	12		X		46832153	
<b>SymbolEnum</b>	Short	2	20		X		1	Note 2
<b>OrderType</b>	Short	2	22		X		LMT	Note 3
<b>SymbolType</b>	Short	2	24		X		SPOT	
<b>BOPrice</b>	Double	8	26		X		51102.5	Note 4
<b>BOSide</b>	Short	2	34		X		BUY	Note 5
<b>BOOrderQty</b>	Double	8	36		X		3.0	
<b>TIF</b>	Short	2	44		X		GTC	Note 6
<b>StopLimitPrice</b>	Double	8	46					
<b>BOSymbol</b>	char[]	12	54		X		BTCUSD	
<b>OrigOrderID</b>	Long	8	66				46832152	Note 9
<b>BOCancelShares</b>	Double	8	74		*			
<b>ExecID</b>	Long	8	82		*			
<b>ExecShares</b>	Double	8	90		*			
<b>RemaingQuantity</b>	Double	8	98					
<b>ExecFee</b>	Double	8	106					
<b>ExpirationDate</b>	char[]	12	114					
<b>TraderID</b>	char[]	6	126					Not used
<b>RejectReason</b>	Short	2	132					
<b>SendingTime</b>	uint64_t	8	134		X			
<b>TradingSessionID</b>	Int	4	142		X		506	

<b>Key</b>	Int	4	146		X		42341	Note 8
<b>DisplaySize</b>	Double	8	150		*			
<b>RefreshSize</b>	Double	8	158		*			
<b>Layers</b>	Short	2	166					
<b>SizeIncrement</b>	Double	8	168					
<b>PriceIncrement</b>	Double	8	176					
<b>PriceOffset</b>	Double	8	184					
<b>BOOrigPrice</b>	Double	8	192				50100.5	Note 10
<b>ExecPrice</b>	Double	8	200					
<b>MsgSeqNum</b>	Long	8	208		X		7948888	
<b>TakeProfitPrice</b>	Double	8	216					
<b>TriggerType</b>	Short	2	224					
<b>Attributes</b>	char[]	12	226		*			Note 7

Note 1: Message types must be valid according to the values listed in the Message Type table above. Since in this example we would like to place a new order on the book, the MsgType field must be set to ORDER\_NEW.

Note 2: Please see previous sections for valid values for the symbol enum.

Note 3: Order types must be a valid value as defined in the OrdType table above. Since in this example we would like to place a limit order on the book, OrdType field should be set to LMT.

Note 4: Price values must be in a price increment according to the value the user received in the BOInstrument message - PriceIncrement field. Example, BTCUSD, symbol enum 1, has a price increment of 0.5. If the user sends a price of 51000.40, this price is invalid since the cents portion of the price is not in a 0.5 increment. The correct price should have been 51000.50 or 51000.00 or 51001.00, all of these are valid values.

Note 5: The valid side fields are:

Enum Name: SIDE	Enum Value	
BUY	1	
SELL	2	

Note 6: TIF valid values:

Enum Name: TIF	Enum Value	
FOK	1	
GTC	2	
IOC	3	
POO	4	
RED	5	
DAY	6	

Note 7: Attributes allow an order to exhibit additional behavior. Currently only the HIDDEN\_ATTRIBUTE and the DISPLAYSIZE\_ATTRIBUTE are available but the other attributes will be available soon. Please see the section above on attribute behavior. In order to set an attribute a user should do something like this:

```
// function accepts the position in the array to set the value and the actual value
BOTransaction.setAttributes(DISPLAYSIZE, 'Y');
```

If the DISPLAYSIZE attribute is set, the DisplaySize and RefreshSize in the message must also be set, if either is not set to a valid value the message will be rejected. If the user sets the HIDDEN\_ATTRIBUTE to 'Y', this order will be hidden.

Note 8: Currently disabled for testing

## 7) Execution/Execution Partial

These two message types are generated when an incoming Quote interacts with a resting order (order already on the book). Upon receiving an Execution message, the order in the Market Data Order Book should be removed. Upon receiving an Execution Partial, the volume of the resting order should be updated to reflect the remaining order quantity.

Field name	Data Type	Data Length	Buffer Offset	Required Field	Required Value	Example Value	Notes
Msg1	Char	1	0	X	T	T	Header
Msg2	Char	1	1				Header
MsgLength	Short	2	2	X	238	238	Header
MessageType	Short	2	4	*		EXECUTION/ EXECUTION_PA RTIAL	
Padding	Short	2	6				Not used
Account	Int	4	8	X		100700	
OrderID	Long	8	12	X		46832153	

<b>SymbolEnum</b>	Short	2	20		X		1
<b>OrderType</b>	Short	2	22		X		LMT
<b>SymbolType</b>	Short	2	24		X		SPOT
<b>BOPrice</b>	Double	8	26		X		51102.5
<b>BOSide</b>	Short	2	34		X		BUY
<b>BOOrderQty</b>	Double	8	36		X		3.0
<b>TIF</b>	Short	2	44		X		GTC
<b>StopLimitPrice</b>	Double	8	46				
<b>BOSymbol</b>	char[]	12	54		X		BTCUSD
<b>OrigOrderID</b>	Long	8	66				46832152
<b>BOCancelShares</b>	Double	8	74				
<b>ExecID</b>	Long	8	82				
<b>ExecShares</b>	Double	8	90				
<b>RemaingQuantity</b>	Double	8	98		X		3.0
<b>ExecFee</b>	Double	8	106				
<b>ExpirationDate</b>	char[]	12	114				
<b>TraderID</b>	char[]	6	126		X		
<b>RejectReason</b>	Short	2	132				
<b>SendingTime</b>	uint64_t	8	134		X		
<b>TradingSessionID</b>	Int	4	142		X		506
<b>Key</b>	Int	4	146		X		42341
<b>DisplaySize</b>	Double	8	150				
<b>RefreshSize</b>	Double	8	158				
<b>Layers</b>	Short	2	166				
<b>SizeIncrement</b>	Double	8	168				
<b>PriceIncrement</b>	Double	8	176				

