



Accelerating Forex Trading to the Next Level

- Superior Quality Order Flow
- Fast Technology
- Innovative Functionality
- Transparency

OUCH PROTOCOL SPECIFICATION DOCUMENT

AutoEx ORDER ENTRY

Revision History		
Version	Last Updated	Updates
1.0	May 15, 2015	
1.0.3	May 20, 2015	Minor text updates for consistency
1.0.4	July 13, 2015	Message offsets are corrected
1.0.5	July 27, 2015	Added 'Reject' message
1.0.6	September 10, 2015	Added MaxDelayOffset and TimeToLive Offset to NEW ORDER EXTENDED message
1.0.7	November 9, 2015	Added reject code 'L' to "Credit limit exceeded available credit"
1.0.8	November 12, 2015	<p>Added the following optional functionality for clients:</p> <p>A client session can be configured to either cancel original order (default) or to keep the original order on a Rejected replace in 'ORDER CXLREPLACE ACK' message</p> <p>Upon request a client session can be configured to return the Original Client Order ID in 'ORDER CANCELED' message</p> <p>If Version set to 2 in LOGIN REQUEST PACKET, Fastmatch will send an extended version of ORDER CXLREPLACE ACK message, which includes CumQty and LeavesQty fields</p>
1.0.10	February 3, 2016	Extended description of MaxDelay and TimeToLive values in NEW ORDER EXTENDED

OUCH Protocol Specification Document	1
AutoEx Order Entry.....	1
Introduction.....	6
1.1 Purpose	6
1.2 Content	6
2 Overview	6
2.1 Scope.....	6
3 Connectivity	6
3.1 FASTMATCH Matching Engine Locations	6
3.2 Connectivity Options.....	6
3.3 Cross-Connectivity	6
3.4 Internet connectivity.....	7
3.5 System/Service Availability	7
3.5.1 Trading Session and STP Session:	7
3.6 Connecting Intraday.....	7
3.7 Protocol Versions	7
3.8 Protocol STACK.....	7
3.9 Supported Message Set	8
3.9.1 Session Level Messages.....	8
3.9.2 Business Messages	10
4 Implementation	10
4.1 Message Encryption.....	10
4.2 Establish Connection / Disconnection	10
4.2.1 Connect	10
4.2.2 Disconnect.....	11
4.3 Ouch Field Types.....	11

4.3.1	Short Field	11
4.3.2	Integer Field	11
4.3.3	Long Field	11
4.3.4	Quantity Field	11
4.3.5	Rate Field.....	11
4.3.6	Timestamp Field	12
4.3.7	Date Field	12
4.3.8	Alphanumeric Field	12
4.4	Orders	12
4.5	Supported Order Types.....	12
4.6	Order Time in force.....	12
4.6.1	Market Order	13
4.6.2	Limit Order	13
4.6.3	Platform minimum order size	13
4.7	Order Execution Notifications.....	13
4.8	Cancel On Disconnect (COD).....	13
5	Session Message Definitions.....	13
5.1	Logical Packet Structure.....	13
5.2	Login Request Packet	14
5.3	Login Accept Packet	14
5.4	Login Reject Packet	14
5.5	Sequenced Data Packet	15
5.6	Server Heartbeat.....	15
5.7	End of Session	15
5.8	Unsequenced Data Packets (Incoming OUCH Messages)	15
5.9	Client Heartbeat.....	16

5.10	Client Logout	16
6	Business Message Definitions	16
6.1	FMouch Header	16
6.2	New Order	17
6.3	New Order Extended	17
6.4	New Order Ack	18
6.5	Cancel Order	18
6.6	Cancel Order Reject	18
6.7	CxlReplace Order Request	18
6.8	Order CxlReplace Ack	19
6.9	Order Canceled	19
6.10	Trade	20
6.11	Reject	20
7	Error Codes	20
7.1	Business error codes	20
7.2	Reject Reason codes	21



INTRODUCTION

1.1 PURPOSE

This document is provided as a guide for FASTMATCH clients, as to how the FASTMATCH's OUCH Protocol (FMOUCH) may be used to establish connectivity with FASTMATCH for the purposes of Foreign Exchange trading. It is intended to provide a guideline for the purpose of order-driven trading of FX and Spot Metals on the FASTMATCH platform using the OUCH protocol. For information about market data please refer to ITCH protocol, available separately.

OUCH is a simplified protocol that allows FASTMATCH participants to enter, replace, and cancel orders and receive executions.

All counterparties will need to certify their trading system with FASTMATCH in the User Acceptance Testing ("UAT") environment before being called production ready.

1.2 CONTENT

Included in this document are the following:

- ❖ General definitions and specifications for clients using OUCH to transmit orders
- ❖ OUCH message formats to be used and details of their expected parameters.

2 OVERVIEW

2.1 SCOPE

The OUCH Interface provides and supports a single connections to place, replace and cancel orders as well as receive execution reports.

3 CONNECTIVITY

3.1 FASTMATCH MATCHING ENGINE LOCATIONS

FASTMATCH matching engine is located in Equinix NY4, LD4 and TY3 Data Centers:

- ❖ NY4, 755 Secaucus Road, Secaucus, NJ 07094
- ❖ LD4, 2 Buckingham Avenue, Slough, Berkshire, SL1 4NB
- ❖ TY3, 1-9-20 Edagawa Koto-Ku Tokyo 135-0051

3.2 CONNECTIVITY OPTIONS

- ❖ Clients have a choice of establishing cross-connect and internet connectivity to FASTMATCH NY4, LD4 and TY3 locations.
- ❖ Local cross-connect to FASTMATCH ECN cages in NY4, LD4 and TY3 data centers could be used for both Production and UAT access.
- ❖ Metro connections from other data centers are accepted.
- ❖ Order Entry traffic will be TCP based with unique target (IP:port) for each OUCH session.
- ❖ No multicast traffic will travel via client connectivity.
- ❖ To start UAT certification process, client can establish the Internet connectivity to FASTMATCH.

3.3 CROSS-CONNECTIVITY

- ❖ All client cross-connect connectivity is 1Gpbs Multimode or Single-mode fiber.

- ❖ FASTMATCH will issue a client a LOA to connect to FASTMATCH ECN with up to two fiber cross-connects.
- ❖ To avoid delays, we ask a client to confirm the correct firm or third-party agent name to be used in LOA.
- ❖ If two cross-connects are ordered, then they will be connected the different access switches for redundancy.
- ❖ BGP is preferred choice even on a single cross-connect connection for support purposes. And static routing is accepted if client hardware cannot support BGP.
- ❖ FASTMATCH will advertise registered IP address space from a registered BGP ASN.
- ❖ Cross-connect will be addressed using RFC 1918 address space (preferred). Registered IP space could be used to avoid the IP address conflict.
- ❖ FASTMATCH will accept client's registered IP Address and BGP ASN. If required, FASTMATCH will assign client's server farm IP addresses and BGP ASN.

3.4 INTERNET CONNECTIVITY

Internet connectivity is available at NY4, LD4 and TY3 locations. There are two internet providers at every location. They may be utilized as main and failover connections. The main connection can be selected based on roundtrip statistics.

3.5 SYSTEM/SERVICE AVAILABILITY

3.5.1 TRADING SESSION AND STP SESSION:

3.5.1.1 SESSION AVAILABILITY:

Market hours for trading are the same as for streaming: the market opens on Sunday at 5:30 PM NY time and closes on Friday at 5:00 PM NY time. The trade service is off-line from 17:00:00 EST/EDT until 17:30:00 EST/EDT, Daily Monday through Thursday. During this time FASTMATCH resets the inbound and outbound sequence numbers on all OUTCH trade sessions.

To successfully connect after 17:30:00 EST/EDT, trading sessions. Typically, when initially logging into a server the client will set the Requested Sequence Number field to 1 and leave the Requested Session field blank in the Login Request Packet

3.6 CONNECTING INTRADAY

For intraday logons, or If the TCP/IP connection is ever broken, the client can then re-log into the server indicating the current session and its next expected sequence number. By doing this, the client is guaranteed to always receive every sequenced message in order, despite TCP/IP connection failures

3.7 PROTOCOL VERSIONS

The only supported version currently is 1.

3.8 PROTOCOL STACK

Fastmatch OUCH specification uses TCP protocol for order submission and execution delivery. Additionally, Fastmatch utilizes SoupBinTCP protocol on top of TCP for session control, reliable delivery, and recovery of execution stream.

SoupBinTCP is a lightweight point-to-point protocol, built on top of TCP/IP sockets that allow delivery of a set of sequenced messages from a server to a client in real-time. SoupBinTCP guarantees that the client receives each message generated by the server in sequence, even across underlying TCP/IP socket connection failures.

SoupBinTCP is ideal for systems where a server needs to deliver a logical stream of sequenced messages to a client in real-time but does not require the same level of guarantees for client generated messages either because the data stream is unidirectional or because the server application generates higher-level sequenced acknowledgments for any important client-generated messages.

SoupBinTCP is designed to be used in conjunction with higher level protocols that specify the contents of the messages that SoupBinTCP messages deliver. The SoupBinTCP protocol layer is opaque to the higher-level messages. Note that unlike the ASCII version, messages may include any possible byte.

SoupBinTCP also includes a simple scheme that allows the server to authenticate the client on login.

Fastmatch employs SoupBinTCP in accordance with [SoupBinTCP, Version 3.0](#) specification with **the two exceptions**:

By default, all integers are sent and received in little-endian format for performance reason.

There is an extra field in Login Request Packet message indicating version number. It is equal 1 for the current version.

This protocol deviation helps to circumvent redundant conversion from little-endian to network order (big-endian) and back, assuming prevalence of Linux/Windows x64-processor based systems. Yet, upon a client's request SoupBinTCP/FMOUCH may be configured to exchange data in network byte order (big-endian) endianness.

3.9 SUPPORTED MESSAGE SET

The supported message set is as follows:

3.9.1 SESSION LEVEL MESSAGES

Direction	Message	Type	Description
In	Login Request Packet	L	The client must send a Login Request Packet immediately upon establishing a new TCP/IP socket connection to the server. Client and server must have mutually agreed upon the username and password fields. They provide simple authentication to prevent a client from inadvertently connecting to the wrong server. Both Username and Password are case-insensitive and should be padded on the right with spaces. The server can terminate an incoming TCP/IP socket if it does not receive a Login Request Packet within a reasonable period of time (typically 30 seconds).
Out	Login Accept	A	Fastmatch server sends a Login Accepted Packet in response to receiving a valid Login Request from the client. This packet

	Packet		will always be the first non-debug packet sent by the server after a successful login request.
Out	Login Reject Packet	J	Fastmatch server sends this packet in response to an invalid Login Request Packet from the client. The server closes the socket connection after sending the Login Reject Packet. The Login Rejected Packet will be the only non-debug packet sent by the server in the case of an unsuccessful login attempt.
Out	Sequenced Data Packet	S	The Sequenced Data Packets act as an envelope to carry the actual sequenced data messages that are transferred from the server to the client. Each Sequenced Data Packet carries one message from the higher-level protocol. The sequence number of each message is implied; the initial sequence number of the first Sequenced Data Packet for a given TCP/IP connection is specified in the Login Accepted Packet and the sequence number increments by 1 for each Sequenced Data Packet transmitted. Since logical packets are carried via TCP/IP sockets, the only way logical packets can be lost is in the event of a TCP/IP socket connection failure. In this case, the client can reconnect to the server and request the next expected sequence number and pick up where it left off.
Out	Server Heartbeat	H	The Fastmatch server should send a Server Heartbeat Packet anytime more than 1 second passes where no data has been sent to the client. The client can then assume that the link is lost if it does not receive anything for an extended period of time.
Out	End of Session	Z	The server will send an End of Session Packet to denote that the current session is finished. The connection will be closed shortly after this packet, and the user will no longer be able to reconnect to the current session.
In	Unsequenced Data Packets	U	The Unsequenced Data Packets act as an envelope to carry the actual data messages that are transferred from the client to the server. These messages are not sequenced and may be lost in the event of a socket failure. The higher-level protocol must be able to handle these lost messages in the case of a TCP/IP socket connection failure.
In	Client Heartbeat	R	The client should send a Client Heartbeat Packet anytime more than 1 second passes where no data has been sent to the server. The server can then assume that the link is lost if it does not receive anything for an extended period of time.

In	Client Logout Request	O	The client may send a Logout Request Packet to request the connection be terminated. Upon receiving a Logout Request Packet, the server will immediately terminate the connection and close the associated TCP/IP socket.
----	------------------------------	---	---

3.9.2 BUSINESS MESSAGES

Direction	Message	Type	Purpose
In	New Order	D	To submit orders to FASTMATCH
In	New Order Extended	E	To submit orders to FASTMATCH with additional instructions
Out	New Order Ack	A	To provide notification on order acceptance
In	Order Cancel Request	F	To cancel an outstanding order or All Open Orders
Out	Order Cancel Reject	R	Response when cancel request cannot be executed
In	Order Cancel/Replace	G	Change the attributes of an order that has been submitted
Out	Order Cancel/Replace Ack	P	To provide notification on order Cancel/Replace acceptance
Out	Order Canceled	C	To provide notification on order cancellation
Out	Trade Execution	T	To provide notification on order fills & partial fills
Out	Reject	J	Reject Message that can be sent in response any message in case server considers message invalid. See Appendix for supported Reject Error codes.

4 IMPLEMENTATION

4.1 MESSAGE ENCRYPTION

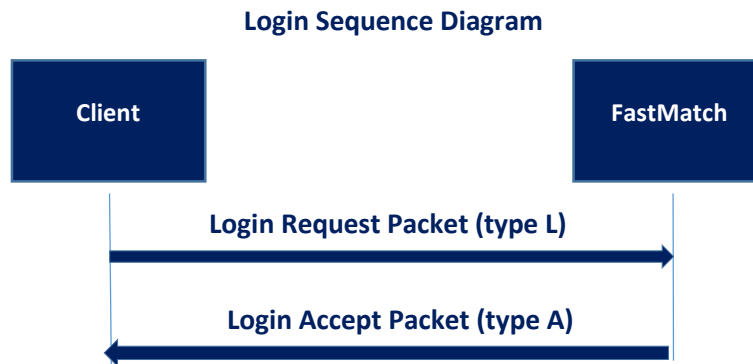
Encryption of binary messages themselves is not supported.

4.2 ESTABLISH CONNECTION / DISCONNECTION

4.2.1 CONNECT

Connection to the system is initiated by the client issuing a Login Request Packet.

If the connection can be accepted, then FASTMATCH will reply with Login Accept Packet, otherwise Login Reject Packet will be sent.



4.2.2 DISCONNECT

Closing of a connection to the system is initiated by sending an End of Session packet to the opposite party. This should then be acknowledged by the other party with a counter Logout message. Either the client or FASTMATCH may initiate disconnect. **All open orders will be canceled on disconnect.** Please see [Cancel On Disconnect](#) section of this document.

4.3 OUCH FIELD TYPES

4.3.1 SHORT FIELD

16bit Signed Integer number

4.3.2 INTEGER FIELD

32bit Signed Integer number. All integers are signed.

4.3.3 LONG FIELD

64bit Signed Integer Number

4.3.4 QUANTITY FIELD

Amounts are using 64bit Long numbers in little-endian format (unless client requests all integers to be in network order). Long value is scaled 2 decimal places to the right (multiplied by 100). For example, amount: 10000.12 should be sent as long number: 1000012. If quantity has no fractional part, it still has to be multiplied by 100. For example: 10000 should be sent as 1000000.

4.3.5 RATE FIELD

Rates are using 32bit Integer number. Integer value is scaled 5 decimal places to the right (multiplied by 100000 (100K)). For example 1.2345 should be sent as long number: 123450. 1200.01 should be sent as 120001000.

4.3.6 TIMESTAMP FIELD

In FMOUCH header, timestamp is a 32bit Integer number, containing number of milliseconds past from 17:00:00 EST/EDT – value date rollover time.

In FMOUCH messages, timestamp field is a 64bit long containing milliseconds from Jan 1, 1970 GMT. Example of such field is TransactTime.

4.3.7 DATE FIELD

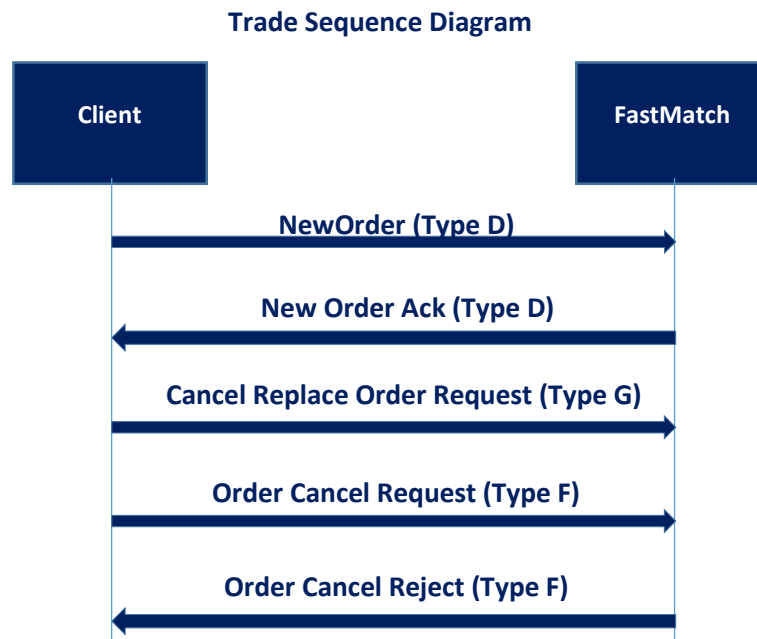
Date is sent as 32bit Integer value containing seconds from Jan 1, 1970 GMT. Examples of such fields are TradeDate and SettlementDate.

4.3.8 ALPHANUMERIC FIELD

Contains alphanumeric characters.

4.4 ORDERS

To place an order the client should submit a New Order message. The following is an example of a trade sequence.



4.5 SUPPORTED ORDER TYPES

The following order types (tag 40) are supported:

- ❖ 1 = Market
- ❖ 2 = Limit

4.6 ORDER TIME IN FORCE

The 'TimeInForce' is used to set the expiration of the order. The following values are supported:

- ❖ 1 - Day (order expires at end of trading session, i.e. 5:00 PM NY time)
- ❖ 2 - IOC, Immediate Or Cancel
- ❖ 3 - FOK, Fill Or Kill

4.6.1 MARKET ORDER

An order type to buy or sell at the current market price. This order is immediately filled at the prevailing market price. Rate field is ignored for market orders.

4.6.2 LIMIT ORDER

At order to buy or sell at a specific price or better.

4.6.3 PLATFORM MINIMUM ORDER SIZE

FASTMATCH has system wide minimum order sizes per currency pair. New orders below the minimum will be rejected. If an existing order falls below the platform minimum due to partial fills, the remaining order quantity will be cancelled. Currently the minimum order sizes for all currency pairs are set to zero. The minimum may change in the future.

4.7 ORDER EXECUTION NOTIFICATIONS

FASTMATCH will send a series of Execution Report messages to the client to provide notification regarding order execution.

Execution Report messages are used to indicate:

- ❖ Order accepted
- ❖ Order partially filled
- ❖ Order filled
- ❖ Order cancelled
- ❖ Order rejected
- ❖ Order replaced

4.8 CANCEL ON DISCONNECT (COD)

Cancel on Disconnect (COD) monitors the ECN system for involuntary lost connections between users and the ECN platform. If a lost connection is detected, COD cancels all resting Orders for the disconnected session for the registered ECN user. It is the user's responsibility to reenter all orders that have been cancelled by COD.

5 SESSION MESSAGE DEFINITIONS

5.1 LOGICAL PACKET STRUCTURE

The SoupBinTCP client and server communicate by exchanging a series of logical packets. Each SoupBinTCP logical packet has:

- A. a two byte little-endian length that indicates the length of rest of the packet (meaning the length of the payload plus the length of the packet type – which is 1)
- B. a single byte header which indicates the packet type
- C. a variable length payload

SoupBinTCP Logical Packet Structure

Two Byte Packet Length	Packet Type	Variable-length payload
------------------------	-------------	-------------------------

For all cases when field length is one byte a corresponding ASCII numeric value of the character can be used. E.g. a printable value 'L' can be represented as one byte decimal 76

5.2 LOGIN REQUEST PACKET

Name	Offset	Length	Value	Description
PacketLength	0	2	Integer	Packet Length - Number of bytes after this field until the next packet
PacketType	2	1	'L'	Log on
Version	3	2	Integer	Version number of the preferred message set
Username	5	6	Alpha	Username
Password	11	10	Alpha	Password
Session	21	10	Alphanumeric	Specifies session that client wants to log into or all blanks to log into current session
NextSeqNum	31	20	Numeric	Next sequence number in ASCII that client wishes to receive or 0 to start receiving most recent message.

5.3 LOGIN ACCEPT PACKET

Fastmatch will send out Login Accepted Packet in response to Login Request from the client.

Name	Offset	Length	Value	Description
PacketLength	0	2	Integer	Packet Length- Number of bytes after this field until the next packet
PacketType	2	1	'A'	Login Accepted
Session	3	10	Alphanumeric	SessionID
SequenceNum	13	20	Numeric	Next sequence number to be sent in ASCII

5.4 LOGIN REJECT PACKET

Name	Offset	Length	Value	Description
PacketLength	0	2	Integer	Number of bytes after this field until the next packet

PacketType	2	1	'J'	Login Rejected
Reject Reason	3	1	Byte	Reject codes: 'A' - Not authorized 'S' - Session not available 'V' - Invalid Version

5.5 SEQUENCED DATA PACKET

Name	Offset	Length	Value	Description
PacketLength	0	2	Integer	Packet Length- Number of bytes after this field until the next packet
PacketType	2	1	'S'	Sequenced data packet
Message	3	Variable	Any	Ouch Outgoing Packet

5.6 SERVER HEARTBEAT

Name	Offset	Length	Value	Description
PacketLength	0	2	Integer	Packet Length- Number of bytes after this field until the next packet
PacketType	2	1	'H'	Server Heartbeat

5.7 END OF SESSION

Name	Offset	Length	Value	Description
PacketLength	0	2	Integer	Packet Length- Number of bytes after this field until the next packet
PacketType	2	1	'Z'	End of Session

5.8 UNSEQUENCED DATA PACKETS (INCOMING OUCH MESSAGES)

Name	Offset	Length	Value	Description
PacketLength	0	2	Integer	Packet Length - Number of bytes after this field until the next packet
PacketType	2	1	"U"	End of Session
Message	3	Variable	Alphanumeric	

5.9 CLIENT HEARTBEAT

Name	Offset	Length	Value	Description
PacketLength	0	2	Integer	Packet Length- Number of bytes after this field until the next packet
PacketType	2	1	'R'	Client Heartbeat

5.10 CLIENT LOGOUT

Name	Offset	Length	Value	Description
PacketLength	0	2	Integer	Packet Length- Number of bytes after this field until the next packet
PacketType	2	1	'O'	Client Logout

6 BUSINESS MESSAGE DEFINITIONS

Business message is transmitted as a 'payload' of the SoupBinTCP packet.

Each business message has a six byte header and a variable length payload. The header has three fields including StreamID, which can be used for logical separation of distinct trading streams within one OUCH session.

Currency pair field in all business messages is 8 byte long, from which the first seven are used.

Currency pairs are formed by concatenating the ISO currency codes (ISO 4217) of the base currency and the counter currency, separating them with a slash character in a CC1/CC2 format. The eighth byte can have any value e.g.

E	U	R	/	U	S	D	anything
---	---	---	---	---	---	---	----------

6.1 FMOUCH HEADER

Field	Offset	Length	Type	Description
MessageType	0	1	Byte	Example: 'D'
Timestamp	1	4	Integer	Time in milliseconds from 17:00pm Eastern Time – value date rollover time
StreamID	5	1	Byte	The ID for the stream (for multi-stream trading only)

6.2 NEW ORDER

MessageType: 'D' Unsequenced

Field	Offset	Length	Type	Description
ClOrderId	6	4	Integer	Client Order ID; it is recommended that ClOrdID be unique throughout the session
CcyPair	10	8	Alpha	Example: CC1/CC2
OrderType	18	1	Byte	'1' – Market, '2' – Limit For Market orders price is ignored
Side	19	1	Byte	'1' – Buy, '2' – Sell
Quantity	20	8	Long	Order Size (x100)
MinQty	28	8	Long	Minimum Order Size (x100)
Rate	36	4	Integer	Order Price (x100000)
TimeInForce	40	1	Byte	'1' – DAY, '2' – IOC, '3' – FOK

6.3 NEW ORDER EXTENDED

(Binary compatible with NewOrder message 'D')

MessageType: 'E' Unsequenced

Field	Offset	Length	Type	Description
ClOrderId	6	4	Integer	Client Order ID
CcyPair	10	8	Alpha	Example: CC1/CC2
OrderType	18	1	Byte	'1' – Market, '2' – Limit For Market orders price is ignored
Side	19	1	Byte	'1' – Buy, '2' – Sell
Quantity	20	8	Long	Order Size (x100)
MinQty	28	8	Long	Minimum Order Size (x100)
Rate	36	4	Integer	Order Price (x100000)
TimeInForce	40	1	Byte	'1' – DAY, '2' – IOC, '3' – FOK
MaxShow	41	8	Long	Iceberg max show
Account	49	8	Alpha	ABC
MaxDelay	57	2	Short	Max LP Fill Time, MaxDelay=0 to interact with hard liquidity only.
TimeToLive	59	4	Integer	Maximum order lifetime and flash-time for IOC/FOK, TTL=0 to not being shown in the market data. If TTL=-1, Fastmatch default values will be used.

6.4 NEW ORDER ACK

MessageType: 'A' Sequenced

Field	Offset	Length	Type	Description
ClOrderId	6	4	Integer	Client Order ID
CcyPair	10	8	Alpha	Example: CC1/CC2
OrderID	18	8	Long	123456
AckStatus	26	1	Byte	'1' – Accept, '2' – Reject
ErrorCode	27	1	Byte	See Full List of Codes

6.5 CANCEL ORDER

MessageType: 'F' Unsequenced

Field	Offset	Length	Type	Description
ClOrderID	6	4	Integer	Client Order ID, 0 – to cancel all active orders
OrigClOrdID	10	4	Integer	Original Order ID
CcyPair	14	8	Alpha	Example: CC1/CC2

6.6 CANCEL ORDER REJECT

MessageType: 'R' Sequenced

Field	Offset	Length	Type	Description
ClOrderID	6	4	Integer	Client Order ID
OrigClOrdID	10	4	Integer	Original Order ID
ErrorCode	14	1	Byte	See list of codes

6.7 CXLREPLACE ORDER REQUEST

MessageType: 'G' Unsequenced

Field	Offset	Length	Type	Description
ClOrderID	6	4	Integer	Client Order ID

OrigClOrdID	10	4	Integer	Original Order ID
CcyPair	14	8	Alpha	Example: CC1/CC2
Quantity	22	8	Long	Order Size (x100)
Rate	30	4	Rate	Order Price (x100000)

6.8 ORDER CXLREPLACE ACK

MessageType: 'P' Sequenced

If Version set to 1 in LOGIN REQUEST PACKET, Fastmatch will send the following ACK message

Field	Offset	Length	Type	Description
ClOrderID	6	4	Integer	Client Order ID
OrigClOrdID	10	4	Integer	Original Order ID
Status	14	1	Byte	'1' – Cancelled, '2' – Replaced '3' - 'Rejected'. A client session can be configured to either cancel original order (default) or to keep the original order on a Rejected replace
ErrorCode	15	1	Byte	See list of codes

If Version set to 2 in LOGIN REQUEST PACKET, Fastmatch will send the following ACK message

Field	Offset	Length	Type	Description
ClOrderID	6	4	Integer	Client Order ID
OrigClOrdID	10	4	Integer	Original Order ID
Status	14	1	Byte	'1' – Cancelled, '2' – Replaced '3' - 'Rejected'. A client session can be configured to either cancel original order (default) or to keep the original order on a Rejected replace
ErrorCode	15	1	Byte	See list of codes
CumQty	16	8	Long	Total number of units filled
LeavesQty	24	8	Long	Amount of units open for further execution. If the Status is Cancelled (in which case the order is no longer active) then LeavesQty could be 0, otherwise LeavesQty = OrderQty - CumQty.

6.9 ORDER CANCELED

MessageType: 'C' Sequenced

Field	Offset	Length	Type	Description
ClOrderID	6	4	Integer	Client Order ID. Upon request a client session can be configured to return the Original Client Order ID

OrderID	10	8	Long	OrderID assigned by ECN
Status	18	1	Byte	'1' – Cancelled by User '2' – Unsolicited ECN Cancel

6.10 TRADE

MessageType: 'T' Sequenced

Field	Offset	Length	Type	Description
ClOrderID	6	4	Integer	Client Order ID
CcyPair	10	8	Alpha	Example: CC1/CC2
FillQty	18	8	Long	1234567 (x100)
FillRate	26	4	Integer	Fill Price (x100000)
Side	30	1	Byte	'1' – Buy, '2' - Sell
ExecID	31	20	Alphanumeric	FX:123456:1234567890
LeavesQty	51	8	Long	Remaining (Unfilled) Qty
Account	59	8	Alpha	Trade Account
LiquidIndicator	67	1	Byte	'A' – Added vsOrder 'B' – Added vs Quote 'R' – Removed vs Order 'S' – Removed vs Quote
ContraCliID	68	2	Short	123
Commission	70	4	Rate	Commission that ECN will collect for Trade (for clients receiving a bill)
TransactTime	74	8	Timestamp	Milliseconds in GMT from Jan 1, 1970
SettlDate	82	4	Date	Seconds in GMT from Jan 1, 1970
TradeDate	86	4	Date	Seconds in GMT from Jan 1, 1970
ContraBroker	90	4	Alpha	FM Contra Broker Codes (See Appendix 8.1)

6.11 REJECT

MessageType: 'J' Sequenced

Field	Offset	Length	Type	Description
MessageType	6	1	Byte	Type of rejected message
RejectCode	7	2	Short	Reject Code
RejectMessage	9	20	Alpha	Reject String

7 ERROR CODES

7.1 BUSINESS ERROR CODES

Error	Code	Description
NoError	0	
DuplicateOrderId	1	Duplicate ClOrdId from client
ExchangeNotOpen	2	Exchange not accepting orders temporarily
InvalidQty	3	Invalid Quantity
InvalidRate	4	Invalid Rate
UnknownStreamId	5	StreamId cannot be mapped to internal clientId
InvalidAccount	6	Invalid account field
UnknownSymbol	7	Symbol not supported
InvalidSide	8	Invalid Side
InvalidMinQty	9	Invalid MinQty (with < 0 or > OrderQty)
UnsupportedTIF	A	Invalid time in force
InvalidMaxShow	B	Invalid max show (< 0 or > OrderQty)
InvalidType	E	Invalid order type
OrderNotFound	F	Order not found by ClOrdId
TooLateToCancel	G	Order already filled or cancelled
CreditLimitExceeded	L	Credit limit exceeded available credit
SymbolDisabled	H	Symbol temporarily disabled for trading
TooManyOpenOrders	I	Client has too many open orders
TradingSuspended	J	Client session is suspended
OtherError	Z	Any other error

7.2 REJECT REASON CODES

Error	Code	Description
NotAuthorized	A	Access is not authorized for used credentials
SessionNotAvailable	S	Session is not available
InvalidVersion	V	Invalid version number