

Omega Securities Inc.  
Omega FIX Interface Specification  
v. 1.0.27  
FIX 4.2

# TABLE OF CONTENTS

REVISION HISTORY .....	2
PURPOSE.....	4
INTRODUCTION .....	5
INTENDED AUDIENCE .....	5
<b>OMEGA SECURITIES INC. INTERFACE SPECIFICATION.....</b>	<b>5</b>
INFORMATION COMMUNICATION AND MESSAGE CONTENT .....	5
<i>Standard Message Header</i> .....	6
<i>Standard Message Trailer</i> .....	7
ADMINISTRATIVE MESSAGES .....	7
APPLICATION MESSAGES.....	7
<i>New Order-Single</i> .....	8
<i>Order Cancel/Replace (Modify) Request</i> .....	12
<i>Order Cancel Request</i> .....	12
<i>Order Cancel Reject</i> .....	13
<i>Crosses</i> .....	13
<i>Order Status Request</i> .....	16
<i>Execution Reports</i> .....	16
<b>ADDENDUM A.....</b>	<b>21</b>
FIX AND THE TRANSPORT LAYER: TCP/IP.....	21
<i>What is TCP/IP?</i> .....	22
<i>TCP/IP Port Connection</i> .....	22
BANDWIDTH REQUIREMENTS .....	23

## Revision History

Date	Version	Description	Author
August 13, 2007	1.0.8	First Public Release	Norman Bates
September 28, 2007	1.0.9	Added FIX Tags required for Compliance and Reporting	NB
October 12, 2007	1.0.10	Drop SymbolSfx(65), now use Symbol dot-suffix notation	NB
October 18, 2007	1.0.11	Added Execution Report ExecRefID(19) to the document	NB
November 19, 2007	1.0.12	Removed ITCH2.0a section, modified ITCH3.0 to 10-char symbols and added UMIRInventoryMatchFlag to the Execution Report	NB
March 22, 2008	1.0.13	Added OnBehalfOf and DeliverTo Tags to Header, Executing Exchange.	NB
August 15, 2008	1.0.14	<ul style="list-style-type: none"> <li>▪ Added Program Trades (6755) to New Order – Single, Program Trades (6755) and Wash Trades (6777) to Execution Reports</li> <li>▪ Changed UMIRInventoryClientMatch (6752) in Execution Reports to tag 6776</li> </ul>	Raymond Tung
September 18, 2008 October 15, 2008	1.0.15	<ul style="list-style-type: none"> <li>▪ In ITCH, added Message Type ‘B’ for Trade Busts.</li> <li>▪ Added SecondaryOrderID (198) to Execution Reports</li> <li>▪ Added Protection (6820) and ProtectionPricImprovement (6821) to New Order – Single</li> <li>▪ Changed Anonymous from Tag 7012 to Tag 6761 in New Order – Single and added tag to Execution Report</li> </ul>	RT
November 3, 2008	1.0.16	<ul style="list-style-type: none"> <li>▪ Added tag SecurityExchange (207) to New Order - Single</li> <li>▪ Added tags SecurityIDSource (22), SecurityID (48), and SecurityExchange (207) to Execution Report</li> </ul>	RT
January 8, 2009	1.0.17	<ul style="list-style-type: none"> <li>▪ Updated ITCH specification with:</li> <li>▪ Starting sequence # in Login Request Packet changed from ‘0’ to ‘1’.</li> <li>▪ Changed the Execbroker length from 4 characters to 3.</li> <li>▪ Changed the Execbroker value from ‘OMG’ to a numerical value.</li> <li>▪ Added a Reserved field with offset 40 and length 1 after the Execbroker field.</li> </ul>	RT
April 20, 2009	1.0.18	<ul style="list-style-type: none"> <li>▪ Correction - For FIX tag OrdStatus (39), added value “9” for suspended</li> <li>▪ Added tag UMIRBypass (6791) to New Order – Single</li> <li>▪ Added tag UMIRBypass (6791) to Execution Report</li> </ul>	RT
January 12, 2010	1.0.20	Remove ITCH and Soup specifications from the document to make this FIX specification only.	RT
July 2, 2010	1.0.21	<ul style="list-style-type: none"> <li>▪ Added notes in description for Tag 100.</li> <li>▪ Added Functional support for Tag 6820, 6821 to comply with order protection rule.</li> </ul>	RT
May 23, 2011	1.0.22	<ul style="list-style-type: none"> <li>▪ Updated values for Tag 30</li> </ul>	RT

July 12, 2012	1.0.23	<ul style="list-style-type: none"> <li>▪ Added ShortMarkingExempt marker (7729)</li> <li>▪ Added Section for Crosses</li> <li>▪ Updated values for Tag 54</li> <li>▪ Updated values for Tag 30</li> <li>▪ Updated values for Tag 207</li> <li>▪ Added Tag 375 for ContraBroker</li> <li>▪ Added Tag 41 to Execution Report (Missing in error)</li> <li>▪ Added Peg Orders</li> </ul>	RT
March 4, 2013	1.0.24	<ul style="list-style-type: none"> <li>▪ Added ActiveOnMarket (6889)</li> <li>▪ Added PostOnMarket (6888)</li> <li>▪ Include references to Lynx ATS</li> </ul>	RT
August 1, 2013	1.0.24	<ul style="list-style-type: none"> <li>▪ Updated description under Visible (9479) tag to specify mid-point pegs are the only hidden order currently supported</li> </ul>	RT
June 10, 2014	1.0.25	<ul style="list-style-type: none"> <li>▪ General cleanup and corrections</li> </ul>	
June 20, 2014	1.0.26	<ul style="list-style-type: none"> <li>▪ Added Tag 12 for Commission</li> <li>▪ Added Tag 13 for CommType</li> </ul>	RT
September 3, 2015	1.0.27	<ul style="list-style-type: none"> <li>▪ Added Tag 7713 for NoTradeFeat</li> <li>▪ Added Tag 7714 for NoTradeKey</li> <li>▪ Added Tag 7715 for NoTradeOrd</li> <li>▪ Added Tag 7716 for NoTradeVol</li> <li>▪ Added Tag 7717 for NoTradePrice</li> <li>▪ Added Tag 7733 for SelfTrade</li> </ul>	RT



## Purpose

The purpose of this Interface Protocol Specification is to provide a reference for participants to develop and interface their trading systems with Omega Securities Inc (Omega, OSI), operating Omega ATS and Lynx ATS.

Subscribers are encouraged to review additional literature and the full text of the Omega Securities Inc Subscriber Manual located at website address: <http://omegaats.com/specifications-policies-and-forms>.

## Introduction

Omega Securities Inc. support the Financial Information eXchange (FIX) Protocol for subscribers to submit order and execute trades on Omega ATS and Lynx ATS.

The FIX Protocol manages and maintains sessions and their state by the exchange of Administrative and Application Messages. The remainder of this document provides the detailed definition of these messages' fields and their formats.

This specification is a supplement to the general FIX Protocol specification, originally developed and currently maintained by the FIX Technical Committee, receiving direction from International Steering Committees, its Global Steering Committee and various Financial Industry Working Groups. The organization's home and additional details on the FIX specification may be found at <http://www.fixprotocol.org>.

## Intended Audience

This document has been developed for the technical audience of system level interface design engineers, software developers and system operation support personnel.

## Omega Securities Inc. Interface Specification

The Omega Securities Inc. Interface FIX Specification is defined herein. This is a working document, the latest and most recent Revision of which is online and may be downloaded from:

<http://www.omegaats.com>

## Information Communication and Message Content

The FIX communication protocol provides for the asynchronous exchange of command and status messages and is implemented in the form of Request and Response packets used for information exchange between clients and the Omega ATS and Lynx ATS servers.

All FIX messages adhere to a general message format specification, in that they consist of:

- Standard Header
- Specialized Content (Request/Response) of an Administrative or Application Message
- Standard Footer

## Standard Message Header

The following fields are supported in the Message Header.

### Standard Message Header

Tag	Field Name	Data Type	Req'd	Comments
8	BeginString	String	Y	FIX.4.2 Note: always unencrypted, this field must be the first field in the message
9	BodyLength	Int	Y	Note: always unencrypted, this field must be the second field in the message
35	MsgType	String	Y	Note: always unencrypted, this field must be the third field in the message
49	SenderCompID	String	Y	Service Provider ID (Subscriber Firm)
56	TargetCompID	String	Y	Assigned receiving firm identifier (Omega Securities Inc.): Always = "OMEG"
34	MsgSeqNum	Int	Y	Unique sequence identification number, unless PossDupFlag = 'Y'
57	TargetSubID	String	N	Assigned value used to identify specific individual or unit intended to receive message. "ADMIN" reserved for administrative messages not intended for a specific user.
115	OnBehalfOfCompID	String	N	Trading partner Company ID used when sending messages via 3 <sup>rd</sup> party.
116	OnBehalfOfSubID	String	N	Trading partner Sub ID used when sending messages via 3 <sup>rd</sup> party.
128	DeliverToCompID	String	N	Trading partner Company ID used when sending messages via 3 <sup>rd</sup> party.
129	DeliverToSubID	String	N	Trading partner Sub ID used when sending messages via 3 <sup>rd</sup> party.
43	PossDupFlag	Boolean	N	Indicates possible retransmission of message with this sequence number: Y = Possible duplicate N = Original transmission
97	PossResend	Boolean	N	Required when message may be duplicate of another message sent under a different sequence number: Y = Possible duplicate N = Original transmission
122	OrigSendingTime	Timestamp	N	Universal Time Coordinate System (UTC)
52	SendingTime	Timestamp	Y	Universal Time Coordinate System (UTC)

## Standard Message Trailer

The following fields are supported in the Message Trailer.

### Standard Message Trailer

Tag	Field Name	Data Type	Req'd	Comments
93	SignatureLength	Int	N	
89	Signature	Data	N	
10	Checksum	String	Y	Three byte, simple checksum (see FIX Protocol Appendix B: CheckSum Calculation for description). ALWAYS LAST FIELD IN MESSAGE; i.e. serves, with the trailing <SOH>, as the end-of-message delimiter. Always defined as three characters. (Always unencrypted)

## Administrative Messages

All FIX 4.2 session level messages are supported by both Omega ATS and Lynx ATS. Refer to [www.fixprotocol.org](http://www.fixprotocol.org) for the specification of these messages and the required tags or optional fields, including values allowed, descriptive meanings and usage notes.

- Logon
- Heartbeat
- Test Request
- Resend Request
- Reject
- Sequence Reset
- Logout

## Application Messages

Omega support FIX application level messages for order entry and execution handling.

Fundamental data-type integrity checking of requests is performed prior to message acceptance. For example, numeric fields must be numbers; the length of fields must be correct, etc. Requests that do not pass fundamental integrity checks or do not contain mandatory tags such as ClOrdID (TAG 11) will be rejected with an Administrative Reject message (MsgType = 3). SessionRejectReason (TAG 373) will contain a Code relevant to the reason for rejection and the text tag (tag 58) may contain a message with additional error detail.

FIX TAGs that are not identified in the following tables will not be processed and are ignored by OmegaATS and Lynx ATS. All required and optional fields specified here shall be treated as follows;



1. The Required fields in request messages will be validated, processed and returned to the caller as specified by the FIX Protocol.
2. All Required and Optional fields specified herein shall be stored by the Omega system and may be utilized by the subscriber reporting sub-system, as required and mutually agreed upon by Omega subscribers and the company.

Omega's interface supports the following FIX Protocol application level message types for Omega ATS and Lynx ATS:

Request Messages

- New Order-Single
- Order Cancel
- Order Cancel/Replace (Modify)
- Order Status

Response Messages

- Order Cancel Reject
- Execution Report

**New Order-Single**

If a required TAG is missing, value is invalid, or value is not supported; an “Execution Report” message will be sent out with ExecTransType = 0 (NEW) and OrdStatus = 8 (REJECTED)

Immediately after Omega receives a New Order-Single, Omega will acknowledge the order by an Execution Report message with ExecTransType = 0 (NEW) and OrdStatus = 0 (NEW).

To comply with the Order Protection Rule, Omega will use tag 100, 6820, and 6821 to protect client orders from booking through or trading through other market places. Tag 100 will be used for routing purposes and tag 6820 and 6821 will be used to protect non routing orders.

Tag 6888 and 6889 will be used to instruct Omega on how an order should interact with Omega ATS and Lynx ATS. Tag 6888 indicates the preferred booking destination if the order or a portion of the order is not immediately executable and needs to be posted to an order book. Tag 6889 indicates whether the order should interact with Omega ATS or Lynx ATS

**New Order-Single Request**

Tag	Field Name	Data Type	Req'd	Comments
	Standard Header		Y	MsgType = D
11	ClOrdID	String	Y	Unique order ID
76	ExecBroker	String	Y	Used for firm identification in third-party transactions. If no third party, then the ID of the primary firm.
6749	UMIRNoMatchId	String	N	This is a legacy tag that is still supported, but

				<p>arrangements should be made to use Tag 7713 and 7714.</p> <p>Orders entered with the same No-Match ID will not be allowed to execute against each other and is effective across Omega ATS and Lynx ATS. The passive side or the active side can be configured to cancel.</p> <p>Note: Values for this field must be assigned by Omega!</p>
7713	NoTradeFeat	String	N	<p>NM = Cancel Newest Order (Active order)  OM = Cancel Oldest Order (Passive order)  DM = Decrement Order (cancels the smaller order, and decreases the larger order by the amount cancelled)  EM = Suppress from Tape (Orders execute, but the print is suppressed from the tape)</p> <ul style="list-style-type: none"> <li>The feature of the active order is considered for two orders having the same NoTradeKey but differing NoTradeFeat values.</li> </ul>
7714	NoTradeKey	String	N	<p>This is a self-assigned alphanumeric key. Orders from the same broker having the same NoTradeKey will exhibit the behaviour specified in Tag 7713 of the active order.</p>
6750	UMIRAccountTypeld	String	N	<p>Required for Canadian Markets regulatory reporting.</p> <p>For Crosses, please use the Buy/Sell Account Type found under the Cross section.</p> <p>NC = Non-Client (Default)  CL = Client  IN = Inventory  MP = ME Pro Order  ST = Equities Specialist  OT = Options Market Maker  OF = Options Firm Account  Note: If unspecified, Default is applied</p>
6751	UMIRUserId	String	Y	<p>Required for Canadian Markets regulatory reporting, the trading system's user id for the trader.</p>
6755	Program Trade	String	N	<p>Order marked as a Program Trade to indicate an offset towards a derivatives trade.</p> <p>Y = Yes  N = No</p>
6757	UMIRJitney	String	N	<p>Order marked as being executed on behalf of another broker, and provided here as their numeric Broker ID. Note: Value is 3-Char Leading Zero-Padded Number.</p> <p>For Crosses, please use the Buy/Sell Jitney found under the Cross section.</p>
6761	Anonymous	Char	N	<p>Flag to indicate if order is anonymous.</p> <p>Y = Yes (Anon is default on Omega)  N = No (Attributed is default on Lynx)</p>
6763	UMIRRegulationId	String	N	<p>Required for Canadian Markets regulatory reporting as</p>

				<p>the Identification marker for orders and trades:</p> <p>For Crosses, please use the Buy/Sell RegID found under the Cross section.</p> <p>NA = Not Applicable (Default)          IA = Insider Account          SS = Significant Shareholder          Note: If unspecified, Default is applied.</p>
6791	UMIRBypass	Char	N	<p>UMIRBypass marker allows an order to execute against only visible liquidity, bypassing hidden liquidity for trade through purposes.</p> <p>Y = Bypass Hidden Liquidity          N = Do not bypass (Default)</p>
6820	Protection	Char	N	<p>(Trade Through/Book Through) Protection marker protects the order against trade throughs and book throughs when entered. If an order will trade through or book through the CBBO, the order can either be price slid to the better price, or rejected as defined by the ProtectionPricImprovement marker, tag 6821          Y = Protection On          N = Protection By-pass          Note: If unspecified, Session default option is applied. Might be Re-price, Cancel or Protection off.</p>
6821	ProtectionPricImprovement	Char	N	<p>Defines whether the order should be re-priced (Y), or cancelled (N) due to Trade Through/Book Through protection          Y = Re-price Order          N = Reject Order          Note: If unspecified, Session default option is applied. Might be Re-price, Cancel or Protection off.</p>
6888	PostOnMarket	String	N	<p>This FIX tag allows the user to specify which of the two markets the order should post on if it is not immediately executable.</p> <p>OMGA = Post on Omega          LYNX = Post on Lynx          Note: If unspecified, Session default option is applied. PostOnMarket has to match ActiveOnMarket</p>
6889	ActiveOnMarket	String	N	<p>This tag specifies which markets the order should interact with and the order of the interaction.</p> <p>O = Omega          L = Lynx          Note: If unspecified, the order will interact with the book it is posted to.          ActiveOnMarket has to match PostOnMarket</p>
109	ClientID	String	N	<p>Note: Echoed on Execution Report if supplied here</p>
1	Account	String	N	<p>Note: Echoed on Execution Report if supplied here.</p> <p>For Crosses, please use the Buy/Sell Account found</p>

				under the Cross section.
21	HandInst	Char	Y	Values: 1 = Private AutoEx, no broker intervention
54	Side	Char	Y	Side of order: 1 = Buy 2 = Sell 5 = Short Sell
7729	ShortMarkingExempt	Char	N	Orders eligible under the IROC definition of SME must be marked accordingly with the values below. 0 = SME
55	Symbol	String	Y	Ticker symbol
207	SecurityExchange	String	N	Exchange code of the listing market. Echoed back in Execution Report if included.
38	OrderQty	Int	Y	Number of shares
40	OrdType	Char	Y	Order Type: 2 = Limit P = Pegged
100	ExDestination	Exchange	N	Destination Exchange – used for order routing via OSI’s Smart Order Router  (Please contact Omega for valid values)
211	PegDifference	Price	N	A price offset which will be used to adjust Visible peg price.
9479	Visible	Char	N	Indicates if the order is visible or hidden. (Currently only mid-point peg orders can be hidden)  Y = Visible N = Hidden
15	Currency	Currency	N	ISO 4217 Currency Code: CAD=Canadian Dollar (Default for CAD traded Symbols) USD=American Dollar (Default for USD traded Symbols) Note: If unspecified, Default is applied
44	Price	Price	Y	Price per share (Limit order price)
59	TimelnForce	Char	N	Specifies duration of order: 0 = Day (Default) 3 = Immediate or Cancel 4 = Fill or Kill Note: If unspecified, Default is applied
111	MaxFloor	Int	N	Maximum number of shares within an order to be displayed on the exchange floor at any given time for iceberg orders.
18	ExecInst	Char	N	Instructions for order handling on exchange trading floor: 0 (zero) = Post on Offer 9 = Post on Bid G = All or none M = Mid-Point Peg P = Market Peg R = Primary Peg

60	TransactTime	Timestamp	Y	Universal Time Coordinate System (UTC)
58	Text	String	N	Free format character string
	Standard Trailer		Y	

Note: The Price and OrderQty Fields (Tag #38 and #44) diverge from the generic FIX protocol, wherein the specification states the fields are optional, and these fields are required by Omega due to its implementation as a “Limit Price” Order System.

### **Order Cancel/Replace (Modify) Request**

Immediately after receiving an Order Cancel/Replace Request Omega ATS or Lynx ATS will acknowledge the request by an Execution Report message referencing ClOrdID with ExecTransType = 0 (New) and OrdStatus = E (Pending cancel/replace).

Omega ATS or Lynx ATS will replace the order referenced in OrigClOrdID and send an Execution Report message referencing ClOrdID with ExecTransType = 0 (New) and OrdStatus = 5 (Replaced).

If a required TAG is missing, value is invalid, value is not supported or the replace failed; an Order Cancel Reject message will be sent referencing ClOrdID. NOTE: the order referenced by the OrigClOrdID will retain its current status on the Omega ATS or Lynx ATS books.

### **Order Cancel/Replace (Modify) Request**

Tag	Field Name	Data Type	Req'd	Comments
	Standard Header		Y	MsgType = G
41	OrigClOrdID	String	Y	Original order id being replaced
ALL FIELDS HERE ARE IDENTICAL TO “NEW ORDER-SINGLE REQUEST”				
	Standard Trailer		Y	

Note: Tag Fields #38 and #44 diverge from the generic FIX protocol, wherein the specification states the fields are optional, and these fields are required by Omega’s vendor specific implementation. With the exception of Price and OrderQty, all fields supplied in this message must match their original fields and as provided by the initial New Order-Single request.

### **Order Cancel Request**

Immediately after receiving an Order Cancel Request Omega ATS or Lynx ATS will acknowledge the request by an Execution Report message referencing ClOrdID with ExecTransType = 0 (New) and OrdStatus = 6 (Pending cancel).

Omega ATS or Lynx ATS will cancel the order and send an Execution Report message referencing ClOrdID with ExecTransType = 0 (New) and OrdStatus = 4 (Canceled).

If a required TAG is missing, value is invalid, value is not supported or the cancel failed; an Order Cancel Reject message will be sent referencing ClOrdID. NOTE: the order referenced by the ClOrdID will retain its current status on the Omega ATS or Lynx ATS books.

### **Order Cancel Request**

Tag	Field Name	Data Type	Req'd	Comments
	Standard Header		Y	MsgType = F
41	OrigClOrdID	String	Y	Original Order ID being cancelled
11	ClOrdID	String	Y	Order id (unique)
54	Side	Char	Y	Side of order: 1 = Buy 2 = Sell 5 = Short Sell
55	Symbol	String	Y	Ticker symbol
60	TransactTime	Timestamp	Y	Universal Time Coordinate System (UTC)
58	Text	String	N	Free format character string
	Standard Trailer		Y	

### Order Cancel Reject

Whenever an Order Cancel/Replace request or Order Cancel request cannot be performed an Order Cancel Reject response will be sent to the client.

#### Order Cancel Reject

Tag	Field Name	Data Type	Req'd	Comments
	Standard Header		Y	MsgType = 9
37	OrderID	String	Y	The client's OrderID (was assigned by OSI).
11	ClOrdID	String	Y	Unique order ID of Order Cancel Request or Order Cancel/Replace Request
41	OrigClOrdID	String	Y	ClOrdID which could not be canceled or cancel/replaced
39	OrdStatus	Char	Y	OrdStatus value after this cancel reject is applied (i.e. current and existing status of the order) Note: see Order Status Request for the list of values
102	CxlRejReason	Char	N	Reason for cancel rejection: 0 = Too late to cancel 1 = Unknown order
434	CxlRejResponseTo	Char	Y	Identifies the type of request that a Cancel Reject is in response to 1 = Order Cancel Request 2 = Order Cancel/Replace Request
58	Text	String	N	Free format character string
	Standard Trailer		Y	

### Crosses

Omega's two order books support Internal Bypass Crosses where a subscriber wishes to cross shares between different accounts within the firm. All crosses must be explicitly marked as bypass, otherwise, the cross will reject. All crosses are automatically attributed unless explicitly specified to be anonymous.

Tag	Field Name	Data Type	Req'd	Comments
	Standard Header		Y	MsgType = D
11	ClOrdID	String	Y	Unique order ID
76	ExecBroker	String	Y	Used for firm identification in third-party transactions. If no third party, then the ID of the primary firm.
6751	UMIRUserid	String	Y	Required for Canadian Markets regulatory reporting, the trading system's user id for the trader.
6761	Anonymous	Char	N	Crosses are automatically attributed unless specifically marked to be anonymous.  Y = Yes N = No (Default)
6791	UMIRBypass	Char	Y	UMIRBypass marker allows an order to execute against only visible liquidity, bypassing hidden liquidity for trade through purposes. This tag must be sent or the cross will reject. Y = Bypass Hidden Liquidity
40	OrdType	Char	Y	Order Type: 2 = Limit
21	HandInst	Char	Y	Values: 1 = Private AutoEx, no broker intervention
44	Price	Price	Y	Price per share (Limit order price)
54	Side	Char	Y	Side of order: 8 = Cross 9 = Cross Short
55	Symbol	String	Y	Ticker symbol
38	OrderQty	Int	Y	Number of shares
59	TimeInForce	Char	N	Specifies duration of order: 0 = Day (Default)
6773	CrossType	Char	Y	I = Internal M = Intentional (Default)
6831	BuyAccount	String	N	Note: Echoed on Execution Report if supplied here
6832	BuyAccountTypeld	String	N	Required for Canadian Markets regulatory reporting. NC = Non-Client (Default) CL = Client IN = Inventory MP = ME Pro Order ST = Equities Specialist OT = Options Market Maker OF = Options Firm Account Note: If unspecified, Default is applied
6833	BuyRegulationId	String	N	Required for Canadian Markets regulatory reporting as the Identification marker for orders and trades: NA=Not Applicable (Default) IA = Insider Account SS = Significant Shareholder Note: If unspecified, Default is applied.
6834	BuyJitney	String	N	Order marked as being executed on behalf of another broker, and provided here as their numeric Broker ID. Note: Value is 3-Char Leading Zero-Padded Number.

6836	SellAccount	String	N	Note: Echoed on Execution Report if supplied here
6837	SellAccountTypeld	String	N	Required for Canadian Markets regulatory reporting. NC = Non-Client (Default) CL = Client IN = Inventory MP = ME Pro Order ST = Equities Specialist OT = Options Market Maker OF = Options Firm Account Note: If unspecified, Default is applied
6838	SellRegulationId	String	N	Required for Canadian Markets regulatory reporting as the Identification marker for orders and trades: NA=Not Applicable (Default) IA = Insider Account SS = Significant Shareholder Note: If unspecified, Default is applied.
6839	SellJitney	String	N	Order marked as being executed on behalf of another broker, and provided here as their numeric Broker ID. Note: Value is 3-Char Leading Zero-Padded Number.
6755	Program Trade	String	N	Order marked as a Program Trade to indicate an offset towards a derivatives trade. Y = Yes N = No
6888	PostOnMarket	String	N	This FIX tag allows the user to specify which of the two markets the cross should be placed on. OMGA = Place on Omega LYNX = Place on Lynx Note: If unspecified, Session default option is applied.
109	ClientID	String	N	Note: Echoed on Execution Report if supplied here
7729	ShortMarkingExempt	Char	N	Orders eligible under the IIROC definition of SME must be marked accordingly with the values below.  1 = Buy Cross SME 2 = Sell Cross SME 3 = Both Buy and Sell Cross SME
207	SecurityExchange	String	N	Exchange code of the listing market. Echoed back in Execution Report if included.
15	Currency	Currency	N	ISO 4217 Currency Code: CAD=Canadian Dollar USD=American Dollar  Note: If unspecified, Symbol trading currency is applied
60	TransactTime	Timestamp	Y	Universal Time Coordinate System (UTC)
58	Text	String	N	Free format character string
	Standard Trailer		Y	



### **Order Status Request**

The client can retrieve the current status of an order with this request. Omega ATS or Lynx ATS will send an Execution Report message referencing ClOrdID with ExecTransType = 3 (Status) and OrdStatus indicating the current status of the order.

If the ClOrdID cannot be found TAG 103 OrdRejReason = 1 (Unknown Order)

The Execution Report OrdStatus will be one of the following:

	<b>OrdStatus</b>
Open	0 (NEW)
Partial	1 (PARTIAL)
Filled	2 (FILL)
Done For Day	3 (DFD)
Canceled	4 (CANCELED)
Replaced	5 (REPLACED)
Pending Cxl	6 (PENDING CANCEL)
Rejected	8 (REJECTED)
Pending Cxl/Rpl	E (PENDING CANCEL/REPLACE)

### **Order Status Request**

Tag	Field Name	Data Type	Req'd	Comments
	Standard Header		Y	MsgType = H
11	ClOrdID	String	Y	Order id (unique)
54	Side	Char	Y	Side of order: 1 = Buy 2 = Sell 5 = Short Sell
55	Symbol	String	Y	Ticker symbol
	Standard Trailer		Y	

### **Execution Reports**

The execution report message will be used to confirm the receipt of an order, changes to an existing order, reject an order request, fill, and/or busted messages. Execution messages are also sent in response to order status requests.

#### TRANSACTION TYPES TAG

The four transaction types (ExecTransType) are:

**0** (New) types indicate that the message represents a new order, a status change of an order, or a new fill against an existing order.

- 1 (Cancel) types indicate that an execution (partial or fill) has been busted.
- 2 (Correct) types indicate details of an execution have been corrected.
- 3 (Status) types indicate the Execution Report is in response to an Order Status Request.

ORDER STATUS TAG

The order status can be determined by examining the ExecTransType and OrdStatus tags. The nine possible order statuses are:

	ExecTransType	OrdStatus
Confirm on New Order – Single	0 (NEW)	0 (NEW)
Execution on an order	0 (NEW)	1 or 2 (PARTIAL or FILL)
Confirmation on Cancel	0 (NEW)	4 (CANCELED)
Confirmation on Cancel/Replace	0 (NEW)	5 (REPLACED)
ACK on Cancel	0 (NEW)	6 (PENDING CANCEL)
Reject on an Order:	0 (NEW)	8 (REJECTED)
ACK on Cancel/Replace	0 (NEW)	E (PENDING CXL/RPL)
Busted Execution	1 (CANCEL)	1 or 2 (PARTIAL or FILL)
Corrected Execution	2 (CORRECT)	1 or 2 (PARTIAL or FILL)

Note: The ExecTransType in the response to an Order Status Request will always be a 3. An Order Status Request cannot be used to determine if an execution has been busted or corrected. See Order Status Request for possible responses.

**Execution Reports**

Tag	Field Name	Data Type	Delivered to Client	Comments
	Standard Header		Y	MsgType = 8
11	ClOrdID	String	Y	Unique order ID
17	ExecID	String	Y	Omega assigned execution identifier
19	ExecRefID	String	N	Required for Cancel and Correct ExecTransType messages, referencing the affected ExecID
76	ExecBroker	String	Y	Used for firm identification in third-party transactions. If no third party, then the ID of the primary firm.
375	ContraBroker	String	N	Used to identify the contra broker on fills. This tag will be missing if the contra is anonymous.
41	OrigClOrdID	String	N	Original Order ID being cancelled
30	LastMkt	String	N	Market Execution of last fills: Alpha - XATS Chi-X - CHIC

				CX2 - XCX2 Lynx - LYNX Omega - OMGA CNSX/Pure - PURE TSX/V - XTSE Select -TMXS
6749	UMIRNoMatchId	String	N	Echoed back if supplied.
7713	NoTradeFeat	String	N	Echoed back if supplied
7714	NoTradeKey	String	N	Echoed back if supplied
7715	NoTradeOrderNum	String	N	This value contains the OSI assigned Order Number of the order that caused a Self Trade Conflict.
7716	NoTradeVol	Int	N	Volume of the prevented trade
7717	NoTradePrice	Price	N	Price of the prevented trade
7733	SelfTrade	Char	N	This tag is sent when a trade is suppressed from tape. Y = Suppressed trade
6776	UMIRInventoryMatchFlag	String	N	Optional tag will be delivered with a value of 'Y' when an order from the same Broker with UMIRAccountTypeld equal to 'CL' matches against another order of the Broker's where UMIRAccountTypeld value is 'IN' to indicate the execution shares apply to Inventory.
6750	UMIRAccountTypeld	String	N	Required for Canadian Markets regulatory reporting. NC = Non-Client (Default) CL = Client IN = Inventory MP = ME Pro Order ST = Equities Specialist OT = Options Market Maker OF = Options Firm Account Note: If unspecified, Default is applied
6751	UMIRUserId	String	Y	Required for Canadian Markets regulatory reporting, the trading system's user id for the trader.
6757	UMIRJitney	String	N	Order marked as being executed on behalf of another broker, and provided here as their numeric Broker ID. Note: Value is 3-Char Leading Zero-Padded Number.
6761	Anonymous	Char	N	Echoed back to indicate if order is anonymous.
6763	UMIRRegulationId	String	N	Echoed back if supplied.
6777	Wash Trade	String	N	Sent back to indicate a wash trade.
6791	UMIRBypass	Char	N	Echoed back to indicate if order is marked to bypass hidden liquidity.
6755	Program Trade	String	N	Echoed back if order was inputted as a Program Trade.
6888	PostOnMarket	String	N	Echoed back if order was inputted with a preferred posting destination.
6889	ActiveOnMarket	String	N	Echoed back if supplied.
9730	TradeLiquidityIndicator	Char	N	Always sent and only valid when LastShares is non-zero and not a cross. This is used to indicate if the shares executed added or removed liquidity from the market: A = Added liquidity R = Removed liquidity

20	ExecTransType	Char	Y	0 = New 1 = Cancel 2 = Correct 3 = Status
39	OrderStatus	Char	Y	0 = New 1 = Partially filled 2 = Filled 3 = Done for day 4 = Canceled 5 = Replaced 6 = Pending cancel/replace 8 = Rejected 9 = Suspended
150	ExecType	Char	Y	0 = New 1 = Partially filled 2 = Filled 3 = Done for day 4 = Canceled 5 = Replaced 6 = Pending cancel/replace 8 = Rejected 9 = Suspended D = Restatement E = Pending Replace
37	OrderID	String	Y	Omega assigned order reference identifier
1	Account	String	N	Account if destination firm requires. Note: Echoed on Execution Report if supplied in New Order-Single request
109	ClientID	String	N	Sending client's ID – will be used for block or institution order Note: Echoed on Execution Report if supplied in New Order-Single request
54	Side	Char	Y	Side of order: 1 = Buy 2 = Sell 5 = Short Sell
55	Symbol	String	Y	Ticker symbol
207	SecurityExchange	String	N	Exchange code of the listing market. Echoed back in Execution Report if included. TO = TSX TV = Venture Exchange CN = CNSX
22	IDSource	String	N	Identifies class of alternative SecurityID  1 = CUSIP
48	SecurityID	String	N	CUSIP identifier of the security
12	Commission	String	N	Displays the per share rate for the execution. Rebates will be displayed as a negative number and charges will be displayed as a positive number.
13	CommType	Char	N	1 = per share
38	OrderQty	Int	Y	Number of shares
40	OrdType	Char	Y	Order Type:

				2 = Limit P = Pegged
15	Currency	Char	Y	ISO 4217 Currency Code: CAD = Canadian Dollar USD = American Dollar Note: If unspecified in the New Order-Single Request, Symbol trading currency will have been applied
44	Price	Price	Y	Limit price (Price per share).
59	TimeInForce	Char	Y	Specifies duration of order: 0 = Day (Default) 3 = Immediate or Cancel 4 = Fill or Kill Note: If unspecified in the New Order-Single Request, Default will have been applied
111	MaxFloor	Int	N	Maximum number of shares within an order to be displayed on the exchange floor at any given time.
103	OrdRejReason	Char	N	Valid value: 0 = Broker/Exchange option 1 = unknown symbol 2 = Exchange closed 5 = Unknown Order 6 = Duplicate Order (e.g. dupe ClOrdID)
18	ExecInst	Char	N	Instructions for order handling on exchange trading floor. (zero) = Post on Offer 9 = Post on Bid G = All or none P = Market Peg R = Primary Peg
211	PegDifference		N	A price offset which can be added or subtracted from the reference price of Visible pegged orders.
60	TransactTime	Timestamp	Y	Universal Time Coordinate System ( UTC)
32	LastShares	Int	N	Qty of shares last bought / sold on this order.
31	LastPx	Price	N	Price of last fill.
14	CumQty	Int	Y	Cumulative Qty. Total number of shares filled.
6	AvgPx	Price	Y	Average price of CumQty shares
198	SecondaryOrderID	String	Y	Assigned by the party which accepts the order. Can be used to provide the OrderID used by an exchange or executing system.
151	LeavesQty	Int	Y	Shares open for further execution
58	Text	String	N	Free format character string
For Crosses				
54	Side	Char	Y	Side of order: 8 = Cross 9 = Cross Short
6773	CrossType	Char	Y	I = Internal M = Intentional (Default)
6831	BuyAccount	String	N	Note: Echoed on Execution Report if supplied here
6832	BuyAccountTypeld	String	N	Required for Canadian Markets regulatory reporting. NC = Non-Client (Default)

				CL = Client IN = Inventory MP = ME Pro Order ST = Equities Specialist OT = Options Market Maker OF = Options Firm Account Note: If unspecified, Default is applied
6833	BuyRegulationId	String	N	Required for Canadian Markets regulatory reporting as the Identification marker for orders and trades: NA=Not applicable (Default) IA = Insider Account SS = Significant Shareholder Note: If unspecified, Default is applied.
6834	BuyJitney	String	N	Order marked as being executed on behalf of another broker, and provided here as their numeric Broker ID. Note: Value is 3-Char Leading Zero-Padded Number.
6836	SellAccount	String	N	Note: Echoed on Execution Report if supplied here
6837	SellAccountTypeld	String	N	Required for Canadian Markets regulatory reporting. NC = Non-Client (Default) CL = Client IN = Inventory MP = ME Pro Order ST = Equities Specialist OT = Options Market Maker OF = Options Firm Account Note: If unspecified, Default is applied
6838	SellRegulationId	String	N	Required for Canadian Markets regulatory reporting as the Identification marker for orders and trades: NA=Not applicable (Default) IA = Insider Account SS = Significant Shareholder Note: If unspecified, Default is applied.
6839	SellJitney	String	N	Order marked as being executed on behalf of another broker, and provided here as their numeric Broker ID. Note: Value is 3-Char Leading Zero-Padded Number.
	Standard Trailer		Y	

Note: Fields with Delivered to Client marked here with “N” may or may not be returned to the client, depending on whether the field has a Default value applied or the client supplies a value for the field. In the cases where either of the two are true (i.e. field has Default value or value supplied by the client) then the field and its value will be Delivered to the Client in the Execution Report.

## Addendum A

### FIX and the transport layer: TCP/IP

In order for a FIX client to interact with a server, it obviously must send and receive data. With the FIX Gateway, data is communicated using TCP/IP, specifically the TCP protocol for an end-to-end stream oriented connection.

## **What is TCP/IP?**

TCP/IP is the standard protocol used in the Internet, as well as used within many corporations for the information technology infrastructure. Originally developed in the late 1960's and early 1970's for the United States Department of Defense, the protocol evolved largely in the academic world until the late 1980's. As the UNIX operating system grew in commercial use (from vendors such as Sun Microsystems, Hewlett-Packard and IBM), TCP/IP began to establish itself as a standard. With the explosion of the Internet, as well as dramatically increased construction of Intranets within corporations, TCP/IP has become the de-facto networking standard for private and public networks throughout much of the world. TCP/IP is actually comprised of 4 protocols: IP, ICMP, UDP and TCP. IP, or Internet Protocol, is a network layer protocol (it sits at layer 3 in the Seven Layer model). It is a data-gram based protocol, with best effort delivery of data. It is only as reliable as the link layer (PPP, Ethernet, FDDI), and depends on transport layer protocols to provide reliable data delivery.

ICMP, or Internet Control Message Protocol, is used for testing and discovery at layer 3. ICMP is basically IP's internal network management protocol and is not intended for use by applications. Two well known exceptions are the ping and traceroute diagnostic utilities:

UDP, User Datagram Protocol (UDP) provides an unreliable packetized data transfer service between endpoints on an internet or network. UDP depends on IP to move packets around the network on its behalf. UDP does not guarantee to actually deliver the data to the destination, nor does it guarantee that data packets will be delivered to the destination in the order in which they were sent by the source

TCP, Transmission Control Protocol provides a reliable byte-stream transfer service between two endpoints on an internet or network. TCP depends on IP to move packets around the network on its behalf. IP is inherently unreliable, so TCP protects against data loss, data corruption, packet reordering and data duplication by adding checksums and sequence numbers to transmitted data and, on the receiving side, sending back packets that acknowledge the receipt of data.

## **TCP/IP Port Connection**

The FIX Client will be required to connect to the FIX Gateway through the TCP/IP layer. Omega will provide the port number to be used to for this connection. The FIX protocol is used by the Omega Book. Subscribers will use the SOUP protocol for session and transport connectivity.



## Bandwidth Requirements

FIX O/E    Min 64KB    Recommended 128KB

Depth of Book    Min 1.5MB    Recommended 3MB