

OMEGA ATS

Omega Alternative Trading System

---

# Omega Interface Specification Guide

OMEGA ALTERNATIVE TRADING SYSTEM

# Interface and Protocol Specifications

**(Version 1.0.22)**

---

Copyright © 2007 All Rights Reserved  
Omega Securities Inc.

# Table of Contents

<b>REVISION HISTORY</b> .....	<b>3</b>
<b>PURPOSE</b> .....	<b>5</b>
<b>INTRODUCTION</b> .....	<b>6</b>
INTENDED AUDIENCE .....	6
<b>OMEGA ATS INTERFACE SPECIFICATION</b> .....	<b>6</b>
INFORMATION COMMUNICATION AND MESSAGE CONTENT .....	6
<i>Standard Message Header</i> .....	6
<i>Standard Message Trailer</i> .....	7
ADMINISTRATIVE MESSAGES .....	8
APPLICATION MESSAGES .....	8
<i>New Order-Single</i> .....	9
<i>Order Cancel/Replace (Modify) Request</i> .....	11
<i>Order Cancel Request</i> .....	12
<i>Order Cancel Reject</i> .....	12
<i>Order Status Request</i> .....	13
<i>Execution Reports</i> .....	13
<b>OMEGA FIX CERTIFICATION GUIDELINES</b> .....	<b>18</b>
<i>New Single Order tests</i> .....	18
<i>Iceberg and Hidden Order tests</i> .....	19
<i>Modify Order tests</i> .....	19
<i>Order Request Reject tests</i> .....	21
<i>Cancel Order tests</i> .....	21
<i>Order Status Message</i> .....	23
<i>Bust and Correction tests</i> .....	23
<i>Done for Day</i> .....	26
<i>Possible Resend</i> .....	27
<i>Session Recovery</i> .....	27
<b>ADDENDUM A</b> .....	<b>29</b>
FIX AND THE TRANSPORT LAYER: TCP/IP .....	29
<i>What is TCP/IP?</i> .....	29
<i>TCP/IP Port Connection</i> .....	29
CONNECTIVITY .....	30
<i>Internet (Various Providers)</i> .....	30
<i>BT Radianz</i> .....	30
<i>Transaction Network Services, Inc. TNS</i> .....	30
<i>Savvis</i> .....	30
BANDWIDTH REQUIREMENTS .....	31

## 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 and MaxFloor to support iceberg and hidden orders.	NB
August 15, 2008	1.0.14	Added Program Trades (6755) to New Order – Single, Program Trades (6755) and Wash Trades (6777) to Execution Reports  Changed UMIRInventoryClientMatch (6752) in Execution Reports to tag 6776	Raymond Tung
September 18, 2008  October 15, 2008	1.0.15	In ITCH, added Message Type ‘B’ for Trade Busts.  Added SecondaryOrderID (198) to Execution Reports Added Protection (6820) and ProtectionPriceImprovement (6821) to New Order – Single Changed Anonymous from Tag 7012 to Tag 6761 in New Order – Single and added tag to Execution Report	RT
November 3, 2008	1.0.16	Added tag SecurityExchange (207) to New Order - Single  Added tags SecurityIDSource (22), SecurityID (48), and SecurityExchange (207) to Execution Report	RT
January 8, 2009	1.0.17	Updated ITCH specification with:  Starting sequence # in Login Request Packet changed from ‘0’ to ‘1’. Changed the Execbroker length from 4 characters to 3. Changed the Execbroker value from ‘OMG’ to a numerical value. Added a Reserved field with offset 40 and length 1 after the Execbroker field.	RT
April 20, 2009	1.0.18	Correction - For FIX tag OrdStatus (39), added value “9” for suspended  Added tag UMIRBypass (6791) to New Order – Single Added tag UMIRBypass (6791) to Execution Report	RT
January 12, 2010	1.0.20	Remove ITCH and Soup specifications from the document to make this FIX specification only.	Greg King
July 2, 2010	1.0.21	Added notes in description for Tag 100. Added Functional support for Tag 6820, 6821 to comply with order	RT

		protection rule.	
October 19, 2010	1.0.22	Updated values for Tag 30	RT

Omega ATS

## **Purpose**

The purpose of this Interface Protocol Specification is that of a Guide for participants and to act as the Outline for test and verification steps, which must be performed for Certification of Interfaces to the Omega ATS System.

Subscribers are encouraged to review additional literature and the full text of the OMEGA ATS Subscriber Manual located at website address: <http://www.omegaats.com/subscribe.php>.

Omega ATS

## Introduction

The Omega ATS exposes a Financial Information eXchange (FIX) interface to subscribers for integrating execution systems with the Omega ATS platform.

The FIX Protocol manages and maintains sessions and their state by the exchange of Administrative and Application Messages and 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 ATS Interface Specification

The Omega ATS 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 at the following location:

<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 server.

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

Version 1.0.22

6

Effective Date: TBD

Tag	Field Name	Data Type	Req'd	Comments
				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 identify (Omega ATS): 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 the Omega 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

The Omega ATS supported FIX application level messages concern 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 a 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 optionally (TAG 58) may contain message Text with additional error detail.

FIX TAGs that are not identified in the following tables will not be processed and are ignored by Omega. 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. Optional fields will not be verified or acted on by Omega.
3. 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 interface supports the following FIX Protocol application level message types:

### 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).

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.

### **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	Orders entered with the same No-Match ID shall not be allowed to execute against one and other. Note: Values for this field must be assigned by Omega!
6750	UMIRAccountTypeId	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.
6755	Program Trade	String	N	Order marked as a Program Trade
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	Flag to indicate if order is anonymous. Y = Yes (Default) N = No
6763	UMIRRegulationId	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.

6791	UMIRBypass	Char	N	UMIRBypass marker allows an order to execute against only visible liquidity, bypassing hidden liquidity for trade through purposes.  Y = Bypass Hidden Liquidity N = Do not bypass (Default)
6820	Protection	Char	N	(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 (Default) N = Protection By-pass
6821	ProtectionPricImprovement	Char	N	Defines whether the order should be re-priced (Y), or rejected (N) due to Trade Through/Book Through protection Y = Re-price Order(Default) N = Reject Order
109	ClientID	String	N	Note: Echoed on Execution Report if supplied here
1	Account	String	N	Note: Echoed on Execution Report if supplied here
21	HandlInst	Char	Y	Values: 1 = Private AutoEx, no broker intervention
54	Side	Char	Y	Side of order: 1 = Buy 2 = Sell 5 = Short Sell 6 = Short Sell Exempt
55	Symbol	String	Y	Ticker symbol
207	SecurityExchange	String	N	Market used to help identify a security.  This field is currently ignored, as Omega will automatically populate this field in the Execution Report to identify which market the symbol is listed on.
38	OrderQty	Int	Y	Number of shares
40	OrdType	Char	Y	Order Type: 2 = Limit
100	ExDestination	Exchange	N	Destination Exchange – used for order routing via Omega’s Smart Order Router  Default = ‘OMG’  (Please contact Omega for valid values)
15	Currency	Currency	N	ISO 4217 Currency Code: CAD=Canadian Dollar (Default) 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.
47	Rule80A	Char	N	Order capacity: I = Customer or Client Order (Default) P = Principal, Agency or Proprietary Note: If unspecified, Default is applied.
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 Note: When post on offer or bid are supplied in this field the setting must be consistent with Side and their use implies "post only" to the Book action.
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 will acknowledge the request by an Execution Report message referencing ClOrdID with ExecTransType = 0 (New) and OrdStatus = E (Pending cancel/replace).

Omega 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 book.

### **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 will acknowledge the request by an Execution Report message referencing ClOrdID with ExecTransType = 0 (New) and OrdStatus = 6 (Pending cancel).

Omega 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 book.

### **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 6 = Short Sell Exempt
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 Omega).
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	

### **Order Status Request**

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

If the ClOrdID can not 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 6 = Short Sell Exempt
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:

	<b>ExecTransType</b>	<b>OrdStatus</b>
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.
30	LastMkt	String	N	Market Execution of last fills:

				Alpha - XATS Chi-X - CHIC Omega - OMGA Pure - PURE TSX - XTSE Venture -XTSX
6749	UMIRNoMatchId	String	N	Orders entered with the same No-Match ID shall not be allowed to execute against one and other. Note: Values for this field must be assigned by Omega!
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	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.
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.
9730	TradeLiquidityIndicator	Char	Y	Only valid when LastShares is non-zero and used to indicate the shares executed did add or remove 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	Describe the type of execution report. Same possible values as OrdStatus.
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 6 = Short Sell Exempt
55	Symbol	String	Y	Ticker symbol
207	SecurityExchange	String	N	Market used to help identify a security. This is populated by Omega. TSE = Toronto Stock Exchange CDX = Toronto Venture Exchange
22	IDSource	String	N	Identifies class of alternative SecurityID  1 = CUSIP
48	SecurityID	String	N	CUSIP identifier of the security
38	OrderQty	Int	Y	Number of shares
40	OrdType	Char	Y	Order Type: 2 = Limit
15	Currency	Char	Y	ISO 4217 Currency Code: CAD = Canadian Dollar (Default) Note: If unspecified in the New Order-Single Request, Default will have been applied
44	Price	Price	Y	Limit price (Price per share).
59	TimelnForce	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.
47	Rule80A	Char	Y	Order capacity: I = Customer Order (Default) P = Principal Note: If unspecified in the New Order-Single Request,

				Default will have been applied
103	OrdRejReason	Char	N	Valid value: 1 = unknown symbol 2 = Exchange closed 5 = Unknown Order 6 = Duplicate Order (e.g. dupe CIOrdID)
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
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
	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.

# Omega FIX Certification Guidelines

The FIX guidelines provide participants with a framework of test messages used to develop a certification script. The script is used to confirm that all message exchanges between the participant and Omega perform as expected. The following tests are for certification to the Omega ATS only.

## New Single Order tests

Order messages have an Order Side, Order Type, Time in Force and Execution Instructions.

### ORDER SIDE

1 Order Sides (Tag 54).						
Test	Date	Market	Purpose		Action	Result
1a		Omega	Buy	54=1	B 100 AZZ LMT Day	ACK --> Fill order
1b		Omega	Sell	54=2	S 1000 BAA LMT Day	ACK -> Partial -> Partial -> Fill
1c		Omega	Sell Short	54=5	SS 100 AZZ LMT Day	ACK -> Partial -> Fill
1d		Omega	Sell Short Exempt	54=6	SP 100 AZZ LMT Day	ACK --> Fill order

### ORDER TYPE

2 Order Types. (Tag 40)						
Test	Date	Market	Purpose		Action	Result
2a		Omega	Market	40=1	B 100 K MKT Day	Reject (35=8, 39=8) Market orders unsupported
2b		Omega	Limit	40=2	S 100 SJ LMT Day	ACK -> Partial -> Partial -> Fill

### TIME IN FORCE

3 Time in Force (Tag 59)						
Test	Date	Market	Purpose		Action	Result
3a		Omega	Day	59=0	B 100 BAA LMT DAY	ACK --> Fill order
3b		Omega	Immediate Or Cancel (IOC)	59=3	B 100 BAA LMT DAY	ACK --> Fill or Cancel or Partial followed by Cxl
3c		Omega	Fill Or Kill (FOK)	59=4	S 100 BAA LMT OPENING	ACK -> Fill or Cxl

### EXECUTION INSTRUCTIONS

4 Execution Instructions (Tag 18)						
Test	Date	Market	Purpose		Action	Result
4a		Omega	AON	18=G	B 100 K LMT DAY	ACK --> Fill
4b		Omega	Post on Bid	18=9	B 100 SJ LMT DAY	ACK --> Fill

4c **Omega** Post on Offer 18=0 B 100 BAA LMT DAY ACK --> Fill  
**Iceberg and Hidden Order tests**

5 MaxFloor (Tag 111)						
Test	Date	Market	Purpose		Action	Result
5a		Omega	Iceberg	111=300	B 3000 BAA LMT DAY	ACK --> Partial 200 shares --> Partial 500 shares --> Fill order
5b		Omega	Iceberg	111=100	B 3000 BAA LMT DAY	Reject --> Displayed quantity < 10%
5c		Omega	Iceberg	111=500	B 400 BAA LMT DAY	ACK --> Full order displayed
5d		Omega	Hidden Order	111=0	B 1000 BAA LMT DAY	ACK --> Partial --> Partial --> Unsolicited Cxl

### Modify Order tests

Tests that insure Order modifications and corrections are exchanged properly. Modify Order is also known as Order Cancel/Replace request.

#### MODIFY ORDERS

6 Modify Order						
Test	Date	Market	Purpose		Action	Result
6a		Omega	Quantity Change - Decrease	35=G & 38=Qty	B 3000 WXX LMT DAY	ACK
					send cxl/rpl to B 2000 WXX LMT DAY	Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
					Receives partial with new OrgOrdQty in tag 38	
					Send cxl/rpl to B 1500 WXX LMT DAY	Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
					Receives partial w/ new OrgOrdQty in tag 38	
						Omega returns --> Fill
					Receives complete fill with final revised OrgOrdQty in tag 38	
6b		Omega	Price Change	35=G & 44=Px	B 3000 BAA LMT DAY	ACK --> Partial
					Send cxl/rpl to lower price	Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
					Receives partial with new Price in tag 44	
					Send cxl/rpl to increase price	Replace Pending (39=6) &

					Replace Accepted (39=5) Omega returns --> Partial
				Receives partial with new Price in tag 44	
					Omega returns --> Fill
				Receives complete fill with final revised Price in tag 44	
<b>6c</b>	<b>Omega</b>	<b>Multiple Corrections followed by FILL</b>	35=G	B 3000 FCC LMT DAY	ACK --> Partial
		Decrease Quantity & Price Change		send cxl/rpl to decrease qty and change price	Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
		Decrease Quantity & Order Type Change		Receives partial with new Price and Quantity Sends cxl/rpl to decrease quantity and change order type	Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
		Decrease Quantity		Receives partial with new OrderType and Quantity Send cxl/rpl to decrease quantity	Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
				Receives partial w/ new Quantity	Omega returns --> Fill
				Receives complete fill with final revised Quantity, Order Type and Price	
<b>6d</b>	<b>Omega</b>	<b>Displayed Quantity Change</b>	35=G & 111=Max Floor	B 4000 BAA LMT DAY Displayed = 600	ACK
				Send cxl/rpl to 111 = 800	Replace Pending (39=6) & (39=5) Omega returns partial
				Send cxl/rpl to decrease order quantity to 1000	Replace Pending (39=6) & (39=5) Omega returns partial
				Send cxl/rpl to decrease order quantity to 500 and 111 = 500	Replace Pending (39=6) & (39=5) Omega returns fill
<b>6e</b>	<b>Omega</b>	<b>Iceberg</b>	35=G & 111=Max Floor	B 3000 BAA LMT DAY	ACK
				Send cxl/rpl to iceberg 111 = 200	Reject --> 111 < 10%
				Send cxl/rpl to iceberg 111 = 500	Replace Pending (39=6) & (39=5) Omega returns Partial 400 shares
				Send cxl/rpl to decrease quantity of order to 200	Reject

6f	Omega	Hidden Order	35=G & 111=Max Floor	Send cxl/rpl to decrease quantity of order to 2200 shares & 111 = 220	Reject --> 111 must equal to board lot
				Send cxl/rpl to decrease quantity of order to 1100 shares & 111 = 100	Reject --> 111 < 10%
				Send cxl/rpl 111 = 0	Replace Pending (39=6) & (39=5) Canceled by trader
				B 1000 BAA LMT DAY 111 = 0	ACK --> Partial
				Send cxl/rpl to decrease quantity of order to 500 shares	Replace Pending (39=6) & (39=5) Omega returns partial
				Send cxl/rpl to 111 = 100	Replace Pending (39=6) & (39=5) Omega returns fill

### Order Request Reject tests

Orders can be rejected for invalid symbol, ambiguous handling instructions, etc. This set of tests ensures the proper exchange of Order Reject (35=8, 39=8) messages.

#### ORDER REJECT

7 Order Reject (Tag 35=8 & 39=8)						
Test	Date	Market	Purpose	Action	Result	
7a		Omega	Reject	35=8. 39=0 & 39=8	Send LMT DAY order with invalid Symbol	ACK --> Omega returns a Reject (35=8, 39=8)
					Verify Reject message	103=1

### Cancel Order tests

This set of tests ensures the proper exchange of Cancel Requests within the resulting Execution Report or Cancel Reject responses.

#### CANCEL EXECUTION REPORTS

8 Cancel Execution Reports						
Test	Date	Market	Purpose	Action	Result	
8a		Omega	Cancel on ACK	B 250 AZZ LMT Day	ACK	
				Send Cancel request	Omega sends Pending CXL (35=8 & 39=6) Omega sends Out (35=8 & 39=4)	
8b		Omega	Cancel on PARTIAL	35=F B 500 FCC LMT DAY	ACK --> Partial --> Partial	
				125=F Send Cancel on Balance	Omega sends Pending CXL (35=8 & 39=6)	

8c	Omega	UnSolicited Cancel on ACK	38=Org Qty	B 200 TZT LMT DAY	Omega sends Cancel Out (35=8 & 39=4) ACK Omega sends unsolicited cancel.
8d	Omega	UnSolicited Cancel on PARTIAL		Verify Balance canceled B 200 TZT LMT DAY	ACK --> Partial --> Partial Omega sends unsolicited cancel.
8e	Omega	Multiple Corrections followed by a CANCEL Decrease Quantity & Price Change	35=G	B 3000 FCC LMT DAY Sends cxl/rpl to decrease qty and change price	ACK --> Partial Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
		Decrease Quantity		Receives partial with new Price and Quantity Send cxl/rpl to decrease quantity	Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
		Decrease Quantity		Receives partial w/ new OrderType and Quantity Send cxl/rpl to decrease quantity	Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
		Cancel Balance	125=F	Receives partial with new Quantity Send Cancel on Balance	Omega sends Pending CXL (35=8 & 39=6) Omega sends Cancel Out (35=8 & 39=4)

CANCEL REJECTS

9 Cancel Rejects (35=9)						
Test	Date	Market	Purpose		Action	Result
9a		Omega	Cancel Reject on ACK	35=9	B 3000 SJ LMT DAY Send Cancel Request	ACK Omega Releases execution. Order gets executed and Cancel rejects/ Too late to cancel (35=9, 102=0)
					Verify order Filled and also receives a REJECT/Reject with Too Late Cancel msg (102=0)	
9b		Omega	Cancel Reject on PARTIAL	35=9	B 3000 WFS LMT DAY	ACK --> Partial --> Partial

				Send Cancel on Balance	Omega Broker Releases execution. Order gets executed and Cancel rejects/ Too late to cancel (35=9, 102=0)
9c	Omega	<b>Cancel Reject on Cxl/Rpl</b>	35=G	Send order and then Cxl/Rpl	Omega Broker Releases execution. Order gets executed and Cancel rejects/ Too late to cancel (35=9, 102=0)
9d	Omega	<b>Cancel/Replace REJECT followed by FILL</b> Decrease Quantity & Order Type Change	35=G	B 3000 K LMT DAY  Send cxl/rpl to decrease qty  Verify Cancel/Replace Request Rejected  Receives fill with the original quantity and LMT price	ACK --> Partial  Replace Pending (39=6) &  Omega returns Cancel/Replace Reject (35=9 & 102=0) TLTC  Omega returns fill for 3000 at LMT price.

### Order Status Message

This test verifies the proper message exchange occurs when the PossResend flag is set on an order request.

#### POSSIBLE RESEND

10 Order Status Message (35=H)					
Test	Date	Market	Purpose	Action	Result
10a		Omega	<b>Order Status on Ack</b>	Send LMT DAY order	ACK
				35=H	Send Order Status Request
10b		Omega	<b>Order Status on Partial</b>	Send LMT DAY order	ACK --> Partial
				35=H	Send Order Status Request

### Bust and Correction tests

This set of tests verifies the exchange of busted executions and Omega quantity and price corrections.

#### BUSTS, PRICE CORRECTIONS AND QUANTITY CHANGES

11 Busts, Price Corrections & Quantity Changes	
--	--

Test	Date	Market	Purpose	Action	Result
11a		Omega	<b>BUST on FILL</b> 20=1 & 39 = 0	B 100 CLL LMT Day  Verify order with 0 CumQty and reopened to ACK state  Receives partial  Verify order with 0 CumQty and reopened to ACK state  Receives complete fill	ACK --> Fill order  Bust fill execution to ACK state  Omega returns --> Partial  Bust partial to ACK state  Omega returns --> Fill
11b		Omega	<b>BUST on Multiple PARTIALS</b> 20=1 & 39=1	B 3000 BAA LMT DAY  Verify order CumQty and Avg Price change  Send Cancel on Balance	ACK -> Partial -> Partial -> Partial --> each w/ different qty and different price Bust 2nd partial execution  Omega sends Pending CXL (35=8 & 39=6) Omega sends Cancel Out (35=8 & 39=4)
11c		Omega	<b>PRICE Correction on FILL</b> 20=2 & 39=2	B 100 TZT LMT Day  Verify order CumQty same and Avg Price changed	ACK --> Fill order  Correct price of fill execution
11d		Omega	<b>PRICE Correction on Multiple PARTIALS</b> 20=2 & 39=1	B 3000 SJ LMT DAY  Verify Avg Price change	ACK -> Partial -> Partial -> Partial --> Fill --> each w/ different qty and different price Change price on 2nd partial execution
11e		Omega	<b>Quantity Change on FILL</b>	B 100 AZZ LMT Day  Verify order is corrected from fill to partial with 50 shares open and 50 filled.	ACK --> Fill order  Correct quantity of fill execution Change executed shares 50 and allow 50 remaining as open.
11f		Omega	<b>BUST &amp; PRICE Change on CXL/RPL'd Order</b> Decrease Quantity	20=1  B 3000 WXX LMT DAY  Send cxl/rpl to decrease qty	ACK -> Partial --> Partial  Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial

				<p>Receives partial with new Quantity Send cxl/rpl to decrease price</p> <p>Change Price</p>	<p>Replace Pending (39=6) &amp; Replace Accepted (39=5) Omega returns --&gt; Partial</p>
				<p>Receives partial w/ new price</p>	<p>Omega BUSTS very first Partial</p>
				<p>Verify Order Quantity, CumQty and Avg Price change</p>	<p>Omega returns --&gt; Partial</p>
				<p>Receives partial</p>	<p>Omega returns PRICE Change on the 2nd Partial</p>
				<p>Verify Order Quantity, CumQty and Avg Price change</p>	
				<p>Send Cancel on Balance</p>	<p>Omega sends Pending CXL (35=8 &amp; 39=6) Omega sends Cancel Out (35=8 &amp; 39=4)</p>
				<p>Verify Order is cancelled</p>	
<b>11g</b>	<b>Omega</b>	<b>TRADE BUST on Cancelled Order</b>	20=1	<p>B 3000 K LMT DAY</p> <p>Verify order is partially filled Send Cancel on Balance</p>	<p>ACK -&gt; Partial -&gt; Partial -&gt; Partial</p>
				<p>Confirm that open shares as working after the bust</p>	<p>Omega sends Pending CXL (35=8 &amp; 39=6) Omega sends Cancel Out (35=8 &amp; 39=4) Omega Busts the very first partial</p>
				<p>Confirm order as cxl'd again and nothing working.</p>	<p>Omega now sends an Unsolicited Cancel</p>
<b>11h</b>	<b>Omega</b>	<b>TRADE AMEND on Cancelled Order</b>	20=2	<p>B 3000 AZZ LMT DAY</p> <p>Verify order is partially filled Sends Cancel on Balance</p>	<p>ACK -&gt; Partial -&gt; Partial -&gt; Partial</p>
				<p>Confirm that the change in price and no shares as reinstated since this order was cxl'd</p>	<p>Omega sends Pending CXL (35=8 &amp; 39=6) Omega sends Cancel Out (35=8 &amp; 39=4) Omega Amends the very first partial</p>

11i	Omega	<b>BUST &amp; PRICE Change on a Canceled order that was CXL/RPL'd</b>	20=1	B 3000 TZT LMT DAY	ACK -> Partial --> Partial
		Decrease Quantity		Send cxl/rpl to decrease qty	Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
		Change Price		Receives partial w/ new Quantity Send cxl/rpl to decrease price	Replace Pending (39=6) & Replace Accepted (39=5) Omega returns --> Partial
				Receives partial with new price Sends Cancel on Balance	Omega sends Pending CXL (35=8 & 39=6) Omega sends Cancel Out (35=8 & 39=4)
				Verify Order is cancelled	Omega returns PRICE Change on the 2nd Partial
				Verify Order Quantity, CumQty and Avg Price change	Omega BUSTS very first Partial and returns a cancel state in the bust
		Verify Order Quantity, CumQty and Avg Price change and that order is still cancelled			

### Done for Day

This set of tests ensures that the Done for Day messages that are sent at market close for all open orders to indicate that the order has been closed by Omega.

#### DONE FOR DAY

12 Done For Day (Tag 35=8 & 39=3)						
Test	Date	Market	Purpose		Action	Result
12a		Omega	Done for Day (DFD)	39=3 & 20=0	Send LMT DAY order	ACK  Omega returns a Done for Day (39=3, 20=0)
					Verify DFD message	
12b		Omega	DFD after Partial	39=3 & 20=0	Send LMT DAY order	ACK  Omega returns a Partial

Omega returns a Done for Day (39=3, 20=0)

Verify Partial followed by DFD message

## Possible Resend

This test verifies the proper message exchange occurs when the PossResend flag is set on an order request.

### POSSIBLE RESEND

13 PossResends (97=Y)						
Test	Date	Market	Purpose	Action	Result	
13a		Omega	PossResend on UnAck'd order	Send LMT DAY order	ACK	
				97=Y & same CIOrdID(11)	ReSend same LMT DAY order	NO ACK
				verify Omega has ONE order and not two		

## Session Recovery

These tests verify the FIX session can reestablish a session that has been disconnected..

### SESSION LEVEL RECOVERY

14 Session Recovery Level Testing						
Test	Date	Market	Purpose	Action	Result	
14a		Omega	REDUCE Incoming SeqNo	<b>Bring DOWN server</b>		
				35=2, 4, A, 0	Reduce Incoming SeqNo by 15 from lower than current and leave Outgoing SeqNo the same	
				Exchange of messages	<b>Bring UP server</b>	
				Logs on. 35=A	logon returned. 35=A	
				Resend Request sent. 35=2	Send resends messages or Sequence reset is given	
				Sequence number is reset and messages process as normal		
				Heartbeat sent		
					Heartbeat returned	

14b

Omega

**DECREASE Incoming SeqNo**

35=2,  
4, A, 0

**Bring DOWN server**

Reduce Incoming SeqNo by 30 from lower than current and leave Outgoing SeqNo the same

Exchange of messages

**Bring UP server**

Logs on. 35=A

logon returned. 35=A

Resend Request sent. 35=2

Resend messages or Sequence reset is given

Sequence # is reset and messages process as normal

Heartbeat sent

Heartbeat returned

14c

Omega

**DECREASE Incoming SeqNo at the same time**

35=2,  
4, A, 0

**Bring DOWN server**

Reduce Incoming SeqNo by 20 from lower than current and increase Outgoing SeqNo by 20

**Bring UP server**

Exchange of messages

Logs on. 35=A

logon returned. 35=A

Resend Request sent. 35=2

Send resends messages or Sequence reset is given

Sequence number is reset and messages process as normal

Heartbeat sent

Heartbeat returned

## **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 ITCH protocol is used by the Omega Book. Subscribers will use the SOUP protocol for session and transport connectivity.

## **Connectivity**

To connect to the Omega ATS platform for FIX Order Execution and Quote Feed Data a wide area network (WAN) or metro area network (MAN) or a Local Area Network (LAN) Internet connection is required. The following vendors are available for MAN and LAN connectivity to the Omega ATS system.

### **Internet (Various Providers)**

IPSEC 3DES VPN Tunnel w/ shared secret

### **BT Radianz**

Dennis Bradley  
Business Development Manager  
BT Radianz  
212-415-4663 Direct  
718-877-2863 Mobile  
dennis.bradley@btradianz.com

### **Transaction Network Services, Inc. TNS**

Carmen Ulloa  
Regional Sales Manager  
TNS, Inc.  
10 S. Riverside Plaza, Suite 1800  
Chicago, Illinois 60606  
312-258-0501  
Fax 312-258-0502

Lisa Shen  
Financial Services Division

703-453-8370 Voice  
703-453-8390 FAX  
lshen@tnsi.com

Cynthia Minardi  
FSD Sales Support (TNS)  
11480 Commerce Park Dr Reston, VA 20191

### **Savvis**

Tony Gerace  
SAVVIS, Inc.  
646.458.4007 direct  
646.244.7962 mobile  
[Anthony.Gerace@savvis.net](mailto:Anthony.Gerace@savvis.net)

## Bandwidth Requirements

FIX O/E Min 64KB Recommended 128B

Depth of Book Min 1.5MB Recommended 3MB

OMEGA ALTERNATIVE TRADING SYSTEM

---