O*U*C*H

Version 1.06

Revised October 21st, 2008

1 Overview

OUCH is the protocol that allows subscribers to enter orders into and receive executions from NASDAQ. It is intended to allow subscribers and their software developers to integrate NASDAQ into their proprietary trading systems or build custom front ends to NASDAQ.

NASDAQ accepts limit orders from system subscribers and executes matching orders against each other. Non-matching orders are added to the NASDAQ Limit Order Book, a database of available limit orders, where they wait to be matched in price-time priority.

OUCH only provides a method for subscribers to send orders to NASDAQ and receive status updates on those orders, it does not convey any information about other subscribers' orders in the book. For a data feed of all orders on the NASDAQ book, please see the ITCH feed specification.

1.1 Architecture

The OUCH protocol is composed of messages passed between the OUCH machine and the client application. Each message ranges in length from 1 to about 100 bytes in length and is composed of non-control ASCII bytes terminated by an ASCII carriage return (13 decimal, 0x0D hex) and optional line feed (10 decimal 0x0A hex). The trailing line feed is always present on messages transmitted outbound from NASDAQ, but is optional on inbound messages. Each Sequenced Outbound Message is assigned a day-unique Sequence Number. The first Sequenced Message sent at the beginning of the day is always the Start of Day Message with a sequence number of 1. The last Sequenced Message is always the End of Day Message.

Each OUCH host machine is assigned a six character Account Name. This Account Name is used during the login and appears on the monthly NASDAQ bill. Each subscriber can have one or more OUCH machines to connect to, and each can have a different Account Name corresponding to a separate office or department.

Every order entered on OUCH is uniquely identified by the combination of the Account Name, User, and Token fields. The User field is an arbitrary four-character field that can be used for anything, but should ideally be used to identify the user who entered the order because NASDAQ monthly bills are broken out based on the User field. The Token field is an arbitrary ten character field that must be day unique for this OUCH Account Name and User. Other than day uniqueness per Account Name and User, there are no other requirements for the Token, so you may wish to use your own system's internal tracking number as the token for convenience.

All numeric fields are right justified, blank padded. Malformed numbers produce undefined results.

All decimal numeric fields are zero filled on the right and blank padded on the left. Malformed decimals produce undefined results.
All alpha fields are left justified, blank padded. Alphanumeric fields can contain the digits

'0' through '9' and the letters "A" through "Z'. Lowercase letters are also permitted and these fields are case sensitive.

1.2 Fault Redundancy

A single Account Name can also be assigned to multiple OUCH machines. These OUCH machines could then be used as mirrors of each other for fault redundancy. In this configuration, both machines would be able to accept orders and any Outbound Sequenced Messages sent would be transmitted by both machines. All un-sequenced messages are machine distinct, so you could log in and out, request rewinds, and enter orders on the two machines independently.

For maximum redundancy, the mirrored machines should be located at geographically diverse data centers with communications carrier access diversity. For example, if a subscriber had two mirrored machines, one could be located at NASDAQ's New York Area data center and one at NASDAQ's Mid-Atlantic data center. The subscriber could increase reliability by using different network carriers to each site. The two lines could also terminate at different subscriber locations on distinct computing platforms.

1.3 Service Bureau Configuration

A single OUCH Account Name can accept orders for one or more firms, allowing a service bureau configuration. The service bureau Account Name must be specifically authorized to enter trades on behalf of each represented subscriber with an NASDAQ Service Bureau Agreement. Once an agreement has been submitted, the Account Name set up as the service bureau may enter orders for the represented firm by putting the represented firm's Market Maker Identifier in the Firm field upon order entry.

2 Outbound Sequenced Messages

Outbound sequenced Messages are transmitted by the OUCH machine to your application. You need to receive every single Outbound Sequenced Message to know where you stand, so there are ways to re-request messages to recover from lost Outbound Sequenced Messages.

2.1 Outbound Sequenced Message Header

All outbound sequenced messages share this common header:

Outbound Sequenced Message Header			
Name	*Len*	*Value*	*Notes*
Type	1	"S"	A single ASCII byte identifying this message as one of the Outbound Sequenced Message types.
Sequence	10	numeric	This day-unique sequence number counts up with each new outbound sequenced message generated. It starts with 1.
Time	5	numeric	This timestamp indicates when this message was originally generated. It is expressed in seconds past midnight., so , for example 9:30:00AM would be $(((9*60)+30)*60)=34200$.

2.1.1 Event Messages

Event messages are outbound sequenced messages that tell you about some predefined event. They look like this:

Event Message Body			
Name	*Len*	*Value*	*Notes*
Outbound Sequenced Header	16	see above	See outbound sequenced message header.
Event Message Identifier	1	"E"	Tells you that this outbound sequenced message is an event message.
Event Specifier	1	alpha	The event specifier tells you which event just happened. See Event Specifiers below.

Event Specifiers	
Specifier	*Event Explanation*
"S"	Start Of Day. This is the first message generated each day and always has a sequence number of one. This tells you that NASDAQ is open and ready for business. It can come anytime from 6:00AM EST to 9:15AM EST, but currently comes at about 8:30AM based on demand.
"E"	End of Day. This is the last message generated. It tells you that NASDAQ is done for the day and we are going home to go to sleep. You should clear out any remaining open orders still in your book and get ready for a brand new day.

2.1.2 Order Messages

Order messages inform you about every event in the lifetime of your orders, from creation to execution to disposition.

Order Message Header			
Name	*Len*	*Value*	*Notes*
Outbound Sequenced Header	16	See above	See Outbound Sequenced Message Header.
Order Message Identifier	1	"O"	Tells you that this Outbound Sequenced Message is an order message.
Order User Identifier	4	alphanum	This is just the User field that was transmitted with the order when entered.
Order Token	10	alphanum	This is just the order Token field as was transmitted with the order when entered.

2.1.2.1 Accepted Order Message

An Accepted Order Message acknowledges the receipt of a valid order. The Accepted Order Message repeats back all of the information originally entered with the order so you can recover from a local database failure or build a redundant remote database of orders on a mirrored machine. The Accepted Order Message also includes a day unique order reference number. Note that if a duplicate User and Token combination is received and all terms of the duped order are identical to the order as it was originally received, the duplicate is ignored.

Accept Order Message			
Name	*Len*	*Value*	*Notes*
Order Message Header	31	see above	See Order Message Header.
Order Action Specifier	1	"A"	Accept Order Action Specifier. Tells that this new order was accepted into NASDAQ.
Order Reference Number	9	numeric	A unique reference number assigned by NASDAQ to the new order. Sequential, but not necessarily continuous. You can safely ignore this, but may wish to database it for problem resolution. It could also be used to cross reference orders into NASDAQ's ITCH data feed.
Buy/Sell Indicator	1	alpha	Buy/sell indicator as entered.
Shares	9	numeric	Total number of shares accepted.

Minimum Shares	9	numeric	Minimum executable shares. This is the smallest number of shares that can be executed from the total order. If Minimum equals Shares, the order is an All or None order. If Minimum equals 0, any number of shares can be executed.
Stock	6	alpha	NASDAQ, NYSE, or AMEX assigned stock symbol.
Price	20	decimal numeric	The limit price of the order. The price appears as a 9-digit whole number portion followed by a period followed by a 10 digit decimal portion. The whole number part is left padded with spaces; the decimal part is right filled with zeros. Example "12.1250000000".
Time in Force	5	numeric	The number of seconds that this order will live before timing out. There are three special values for Time in Force. 99997 indicates that the order was converted to an IOC order (equivalent to time in force = 0) because it would have locked or crossed the NASDAQ market if it was left on the NASDAQ book. 99998 indicates that the order will live until the market close on the primary market, and 99999 indicates that the order will live until the end of the NASDAQ trading day.
Firm	4	alpha	The NASDAQ Market Maker Identifier for the order entry firm.
Capacity	1	alpha	Can be "P" for principal, "A" for agency, or "R" for riskless. For reporting purposes only.
Display	1	alpha	"A" = Attributable-Price to Display "Y" = Anonymous-Price to Comply "N" = Non-Display "Z" = Entered as displayed but changed to non-displayed (Priced to comply)
Reserved for NASDAQ	1	alpha	Blank

2.1.2.2 Canceled Order Message

A Canceled Order Message informs you that an order has been reduced or canceled. This could be acknowledging a Cancel Order Request or it could be as a result of the order timing out. Note that a Cancel Order Message does not necessarily mean the entire order is dead, some portion of the order may still be alive.

Canceled Order Message			
Name	*Len*	*Value*	*Notes*
Order Message Header	31	see above	See Order Message Header.
Order Action Specifier	1	"C"	Canceled Order Action Specifier. Tells you that this order has been reduced or canceled.
Decrement Shares	9	numeric	The number of shares decremented from the order. This could be any number less than or equal to the number of shares previously left in the order. This number is incremental, not cumulative.
Remaining Shares	9	numeric	The number of shares currently left alive in this order. Sent only as a reference for double-checking. If this is zero, the order is dead.
Reason	4	alpha	The reason the order was reduced or canceled. Reasons include User Cancel, Timeout, and Halt. See Reason Specifiers below.

Reduce Order Reason Specifiers	
Specifier	*Event Explanation*
"#USR"	User requested cancel. You previously sent a Cancel Request Message on all or part of this order.
"#IOC"	Immediate or Cancel order. This order was originally sent with a timeout of zero, and no further matches were available on the book, so these remaining shares were immediately canceled out.
"#TME"	Timeout. This specifier lets you know that this order has timed out after the number of seconds specified in the Time in Force field elapsed.
"#HLT"	Halt. The stock was halted on its primary market so NASDAQ automatically canceled any open orders in that stock. The number of shares in the Decrement Shares field should exactly equal the number of pending shares left in the order. This order is now dead.
"#SUP"	Supervisory. This order was manually canceled or reduced by an NASDAQ supervisory terminal. This is usually in response to a user requested purge via telephone.

"#LCK"	This order was automatically immediately canceled because, if it were added to the NASDAQ book, it would have locked or crossed the primary inside market. This Reason only occurs during market hours. Outside market hours, you are free to lock or cross the primary market.
"#DNT"	This order was automatically canceled because it was a potential short sale tick violation.

2.2 Outbound Reject Messages

Outbound Reject Messages tell you that an Enter Order or Cancel Order Inbound Message was rejected. These messages are not sequenced.

2.2.1.1 Rejected Order Message

A Rejected Order Message may be sent in response to an Enter Order Message if for some reason the order can not be entered. The reason for the rejection is given. Since the order was never accepted, the order is dead. No further actions are permitted on this order. Unless the reason for rejection was Duped Token, the token of the rejected order may be reused if desired.

Your application should handle unknown Rejected Order Reason Specifiers resiliently and alert with the unknown reason.

Rejected Order Message			
Name	*Len*	*Value*	*Notes*
Type	1	"J"	A single ASCII byte identifying this as an Outbound Reject Message.
Reject Specifier	1	"O"	Rejected Order specifier. Tells you that this is a reject of an order sent using the Enter Order Message.
Order User	4	alphanum	This is just the User field that was transmitted with the order when entered.
Order Token	10	alphanum	This is just the order Token field as was transmitted with the order when entered.
Reason	8	alpha	The reason the order was rejected. Left justified, blank filled.

Rejected Order Reason Specifiers	
Specifier	*Event Explanation*
"DUPETOKN"	Duped Token. The specified token was already used today for this Account Name and user. Try again with a fresh token.
"TESTMODE"	This Account Name is configured for test mode and is not able to accept orders in non-TEST securities.
"HALT"	There is currently a trading halt on the primary exchange, no orders can be accepted.
"MMID"	This Account Name is not authorized to accept orders for the specified firm.

"BUYSELL"	The Buy Sell field must be "B", "S", "T", or "E".
"SHARES"	The Shares field must be all numeric.
"MINIMUM"	The Minimum field must be all numeric and less than or equal to the Shares field.
"THRSHOLD"	The number of shares entered must be less than the safety threshold on this Account. The subscriber can change the threshold at any time.
"STOCK"	The Stock field must be a valid issue, tradable on NASDAQ.
"PRICE"	The Price filed must be a well-formed numeric.
"PA"	The Principal/Agency field must be "P" or "A".
"DISPLAY"	The Displayed field must be "A", "Y" or "N".
"FIRM"	The Firm entered must be authorized for this account name. To authorize additional firms, use the NASDAQ Service Bureau Agreement.
"TIF"	The TIF entered must be a positive number.
"DOWNTICK"	This order was a prohibited short sale in a listed security at or below the last trade price on a downtick.
"NOQSR"	This order was entered by a firm currently setup to clear using ACT but NASDAQ's ACT System has already closed for the day. Call NASDAQ Subscriber Services Department to get your firm setup to clear using QSR, which allows you to enter trades whenever NASDAQ is open.
"MASTER"	This order was rejected because of a problem with NASDAQ's security master system. The symbol, inside market, and tick information could not be obtained so the order was rejected. You should immediately call NASDAQ's subscriber support and inform them of the reject code.

2.2.1.2 Reject Cancel Message

A Reject Cancel Message is sent in response to a Cancel Order Message that could not be honored. The reason for the rejection is given. Note that there is no response to a too late to cancel condition. If a Cancel Order Message is sent for an order that has already been fully canceled or executed, it will be ignored. A Reject Cancel Message is only sent if you try to cancel an unknown order.

Reject Cancel Message			
Name	*Len*	*Value*	*Notes*
Type	1	"J"	Identifies this as an Outbound Reject Message.
Reject Specifier	1	"K"	Reject Cancel Specifier. Indicates that a Cancel Order Message was rejected.
Order User	4	alphanum	The User field that was transmitted with the order when entered.
Order Token	10	alphanum	This is just the order Token field as was transmitted with the order when entered.
Reason	4	alpha	The reason the order could not be reduced or canceled. Left justified, blank filled.

Can Not Cancel Reason Specifiers	
Specifier	*Event Explanation*
"#UNK"	The requested order, as specified by the User and Token fields, was never accepted for this Account Name today.

2.1.2.3 Executed Order Message

An Executed Order Message informs you that all or part of an order has been executed.

Executed Order Message			
Name	*Len*	*Value*	*Notes*
Order Message Header	31	see above	See Order Message Header.
Order Action Specifier	1	"E"	Executed Order Action Specifier. Tells you that all or some of this order has been executed.
Executed Shares	9	numeric	Incremental number of shares just executed. This could be any number less than or equal to the number of shares previously left in the order.

Remaining Shares	9	numeric	The number of shares currently left alive in this order. Sent only as a reference for double-checking. If this is zero, the order is dead.
Execution Price	20	decimal numeric	The price at witch these shares were executed. The execution price will always be equal to or better than the entry price. The price appears as a 9-digit whole number portion followed by a period followed by a 10 digit decimal portion. The whole number part is left padded with spaces; the decimal part is right filled with zeros. Example "12.1250000000".
Reserved for NASDAQ	4	alpha	blank
Liquidity Flag	1	alpha	See Liquidity Flag Table below
Match Number	9	numeric	The NASDAQ assigned Match Number. Each match of a buy and a sell order on NASDAQ is assigned a unique match number

Liquidity Flags	
Flag	*Value*
A	Added
R	Removed
O	Opening Cross (billable)
M	Opening Cross (non-billable)
C	Closing Cross (billable)
L	Closing Cross (non-billable)
H	Halt/IPO Cross (billable)
K	Halt/IPO Cross (non-billable)
I	Intraday/Post-Market Cross
J	Non-displayed and added liquidity

2.1.2.4 Broken Trade Message

A Broken Trade Message informs you that a previously executed trade has been broken. The trade is no longer good and will not clear. The reason for the break is given.

Broken Trade Message			
Name	*Length*	*Value*	*Notes*
Order Message Header	31	see above	See Order Message Header.
Order Action Specifier	1	"B"	Broken Trade Specifier. Tells you that this order had been executed but is now broken.
Reason	4	see below	The reason the trade was broken. Left justified, blank filled.
Match	9	numeric	Uniquely identifies the match that was broken.

Broken Trade Reason Specifiers	
Specifier	*Event Explanation*
"#ERR"	The trade was deemed clearly erroneous pursuant to NASDAQ's administrative procedure.
"#SYS"	The trade was broken in response to a system problem.
"#CON"	By mutual consent. The two parties agreed to break the trade.
"#SUP"	The trade was manually broken by an NASDAQ supervisory terminal.
"#EXT"	Trade was broken by an external influence. This may be a regulator or another marketplace.

2.2 Outbound Reject Messages

Outbound Reject Messages tell you that an Enter Order or Cancel Order Inbound Message was rejected. These messages are not sequenced.

2.2.1.1 Rejected Order Message

A Rejected Order Message may be sent in response to an Enter Order Message if for some reason the order can not be entered. The reason for the rejection is given. Since the order was never accepted, the order is dead. No further actions are permitted on this order. Unless the reason for rejection was Duped Token, the token of the rejected order may be reused if desired.

Your application should handle unknown Rejected Order Reason Specifiers resiliently and alert with the unknown reason.

Rejected Order Message			
Name	*Len*	*Value*	*Notes*
Type	1	"J"	A single ASCII byte identifying this as an Outbound Reject Message.

Reject Specifier	1	"O"	Rejected Order specifier. Tells you that this is a reject of an order sent using the Enter Order Message.
Order User	4	alphanum	This is just the User field that was transmitted with the order when entered.
Order Token	10	alphanum	This is just the order Token field as was transmitted with the order when entered.
Reason	8	alpha	The reason the order was rejected. Left justified, blank filled.

Rejected Order Reason Specifiers	
Specifier	*Event Explanation*
"DUPETOKN"	Duped Token. The specified token was already used today for this Account Name and user. Try again with a fresh token.
"TESTMODE"	This Account Name is configured for test mode and is not able to accept orders in non-TEST securities.
"HALT"	There is currently a trading halt on the primary exchange, no orders can be accepted.
"MMID"	This Account Name is not authorized to accept orders for the specified firm.
"BUYSELL"	The Buy Sell field must be "B", "S", "T", or "E".
"SHARES"	The Shares field must be all numeric.
"MINIMUM"	The Minimum field must be all numeric and less than or equal to the Shares field.
"THRSHOLD"	The number of shares entered must be less than the safety threshold on this Account. The subscriber can change the threshold at any time.
"STOCK"	The Stock field must be a valid issue, tradable on NASDAQ.
"PRICE"	The Price filed must be a well-formed numeric.
"PA"	The Principal/Agency field must be "P" or "A".
"DISPLAY"	The Displayed field must be "Y" or "N".
"FIRM"	The Firm entered must be authorized for this account name. To authorize additional firms, use the NASDAQ Service Bureau Agreement.
"TIF"	The TIF entered must be a positive number.
"DOWNTICK"	This order was a prohibited short sale in a listed security at or below the last trade price on a downtick.
"NOQSR"	This order was entered by a firm currently setup to clear using ACT but NASDAQ's ACT System has already closed for the day. Call NASDAQ Subscriber Services Department to get your firm setup to clear using QSR, which allows you to enter trades whenever NASDAQ is open.

Deleted: "A",

"MASTER"	This order was rejected because of a problem with NASDAQ's security master system. The symbol, inside market, and tick information could not be obtained so the order was rejected. You should immediately call NASDAQ's subscriber support and inform them of the reject code.

2.2.1.2 Reject Cancel Message

A Reject Cancel Message is sent in response to a Cancel Order Message that could not be honored. The reason for the rejection is given. Note that there is no response to a too late to cancel condition. If a Cancel Order Message is sent for an order that has already been fully canceled or executed, it will be ignored. A Reject Cancel Message is only sent if you try to cancel an unknown order.

Reject Cancel Message			
Name	*Len*	*Value*	*Notes*
Type	1	"J"	Identifies this as an Outbound Reject Message.
Reject Specifier	1	"K"	Reject Cancel Specifier. Indicates that a Cancel Order Message was rejected.
Order User	4	alphanum	The User field that was transmitted with the order when entered.
Order Token	10	alphanum	This is just the order Token field as was transmitted with the order when entered.
Reason	4	alpha	The reason the order could not be reduced or canceled. Left justified, blank filled.

Can Not Cancel Reason Specifiers	
Specifier	*Event Explanation*
"#UNK"	The requested order, as specified by the User and Token fields, was never accepted for this Account Name today.

2.3 Heartbeat Message

A Heartbeat Message is transmitted about once every 15 seconds during periods of inactivity. The Heartbeat Message's primary purpose is to periodically let you know the expected sequence number so you can detect missed messages even during times of low message traffic levels. Seeing the frequent Heartbeat Messages can also reassure your application of link integrity during times when there is no message traffic.

Any Heartbeat messages sent during an active rewind will indicate the expected sequence number of the next Outbound Sequenced Message to be transmitted to avoid triggering redundant rewind requests.

You must respond to a Heartbeat Message by immediately transmitting a Heart Beat Response Message. If you fail to respond to four consecutive Heartbeat Messages, you will receive a Goodbye Message and be disconnected.
The Time field in the Heartbeat Message is based on the local clock of the machine you are connected to. The Time field in Outbound Sequenced Messages is based on the NASDAQ global clock. Time values are expressed in seconds past midnight., so , for example 9:30:00AM would be $(((9*60)+30)*60)=34200$.

Heartbeat Message Format			
Name	*Len*	*Value*	*Notes*
Type	1	"H"	Tells you that this is indeed a Heartbeat Message.
Sequence	10	numeric	This is the expected Sequence Number of the next Outbound Sequenced Message to be transmitted.
Time	5	numeric	This timestamp indicates when this Heartbeat Message was generated. Because Heartbeat Messages are not buffered or queued, this should be the exact time less transmission delays.

3 Inbound Messages

Inbound messages are sent from your application to the OUCH machine. They are not sequenced. Inbound Messages may be repeated benignly. This gives you the ability to re-send any Inbound message if you are not sure whether it was received by your OUCH machine.

The idea of benign inbound message retransmission with end-to-end acknowledgement is fundamental to NASDAQ's fail-over redundancy. Should your connection ever fail, there is no way to know what pending messages actually made it over the link before failure. A robust OUCH client can safely re-send any pending messages over a mirrored linked with worrying about generating duplicates. This applies to NASDAQ's disaster fail over capability as well, should NASDAQ ever need to fail over to the backup site, some messages sent at the moment of the failure may be lost. A robust application would simply re-send the pending messages, making the fail over seamless to the end user.

For example, if your line goes down immediately after you send an Enter Order Message, you can safely re-send the message to any OUCH machine assigned to the same Account Name without worrying about duping the order. You could even technically send the same order down multiple lines simultaneously, although it's not recommended. If you ever feel like you've waited too long for a response to any Inbound Message, just re-send it.

The retransmission timeout is dependent primarily on link latency. For TCP/IP over a 56K frame relay connection, 1/2 second is probably a good benchmark, but we will let you know if your connection starts generating excessive redundant messages.

That said, if you want to reduce the complexity of your application, you can ignore the entire retransmission logic altogether. The need for a retransmission is very rare over point-to-point TCP/IP connections and 99.99% may be good enough for your application. In many OUCH implementations, it may be appropriate to not display the order until you receive the Accept Order Message, so it's left up to the user to re-enter a new order if he

sees that his order has not been accepted within the expected period of time. It's up to you.

3.1 Enter Order Message

The Enter Order Message lets you enter a new order into NASDAQ. Each new order must have a day unique User and Token combination for this Account Name. If you send a valid order, you should receive an Accepted Order Message. If you re-send an order with the same User and Token and identical terms, the duplicate order will be ignored.

			Enter Order Message
Name	*Length*	*Value*	*Notes*
Type	1	"O"	Identifies this message an Enter Order Message type.
Order User	4	alphanum	This is a freeform alphanumeric field. You can place any information you like here with the restriction that the combination of the User plus the Token must be day unique for each order entered.
Order Token	10	alphanum	This is a freeform alphanumeric field. You can place any information you like here with the restriction that the combination of the User plus the Token must be day unique for each order entered.
Buy/Sell Indicator	1	alpha	"B" = buy order "S" = sell order "T" = sell short, client affirms ability to borrow securities in good deliverable form for delivery within three business days "E" = sell short, client affirms ability to borrow securities in good deliverable form for delivery within three business days "U" = short sale, client unable to affirm delivery of securities within three business days
Shares	9	numeric	Total number of shares.
Minimum	9	numeric	Minimum executable shares. This is the smallest number of shares that can be executed from the total order. If Minimum equals Shares, the order is an All or None order. If Minimum equals 0, any number of shares can be executed. Must be less than or equal to Shares.

16

			If the Minimum Shares field is greater than 0 then the Time In Force must be 0.
Stock	6	alpha	NASDAQ, NYSE, or AMEX assigned stock symbol.
Price	20	decimal numeric	The limit price of the order. The price is a 9-digit whole number portion followed by a period followed by a 10 digit decimal portion. The whole number part must be left padded with spaces and the decimal part must be right filled with zeros. Example "12.1250000000"
Time in Force	5	numeric	The number of seconds that this order will live before being timed out. There are two special values for Time in Force, 99998 indicates that the order should live until the market close on the primary market, and 99999 indicates that the order should live until the end of the NASDAQ trading day. A Time In Force of 0 indicates an Immediate or Cancel order.
Firm	4	alpha	This field should contain the all caps NASDAQ Market Maker Identifier for the order entry firm. One Account Name can enter orders for multiple firms in a Service Bureau configuration.
Capacity	1	alpha	Can be "P" for principal, "A" for agency, or "R" for riskless. Used for reporting purposes only.
Display	1	alpha	"Y" = Anonymous - Price to Comply "N" = Non-Display
Reserved for NASDAQ	1	alpha	Blank

It is the subscriber's responsibility to ensure compliance with the SEC Order Handling Rules when using Non-Display orders. These orders do not qualify for the SEC Display

Alternative Exemption.

3.2 Cancel Order Message

The Cancel Order Message is used to request that a previously entered order be canceled or reduced. In the Cancel Order Message, you must specify the number of shares you wish to have remaining after the cancel. To cancel the entire order, you would enter a Cancel Order Message with a Shares field of zero.

This allows redundant Cancel Order Messages to be sent benignly.

Cancel Order Message			
Name	*Len*	*Value*	*Notes*
Type	1	"X"	A single byte identifying this message as a Cancel Order Message.
Order User	4	alphanum	This is the User I.D. that was transmitted with the order when entered.
Order Token	10	alphanum	This is the order token as was transmitted with the order when entered.
Shares	9	numeric	This is the number of shares to reduce to. Entering a zero here will cancel any remaining open shares on this order.

3.3 Inbound Link Control Messages

Link Control Messages do not affect the current outbound sequence number.

3.3.1 Login Request Message

The first thing you should do upon establishing a link layer connection to an OUCH machine is to request a login with a Login Request Message. Both the Account Name and Password fields will be assigned to you when you request an OUCH connection, both are case insensitive and may only contain alphanumerics.

If you do not log in within 30 seconds after establishing a connection, your connection will be terminated.

You can re-log in even if you are currently logged in.

After you send a Login Request, you should wait for either a Welcome Message or a Goodbye Message and react accordingly.

Login Request Message Format			
Name	*Len*	*Value*	*Notes*
Type	1	"L"	Tells the OUCH machine that this is a Login Request Message.
Account Name	6	alphanum	This is your assigned OUCH Account Name. It is case insensitive. You must fill with spaced on the right.
Password	10	alphanum	This is your assigned OUCH Password. It is case insensitive. You must fill with spaces on the right

3.3.2 Logout Request Message

After you send a Logout Request Message, you will receive a Goodbye Message. It's a good idea to send a Logout Request Message before terminating your application. Your OUCH machine will send a GoodBye Message and terminate your TCP/IP connection after receiving a Logout Request Message.

Logout Request Message Format			
Name	*Len*	*Value*	*Notes*
Type	1	"F"	Tells the OUCH machine that this is a Logout Request Message.

3.3.3 Heartbeat Response Message

You must respond to each Heartbeat Message you receive by immediately transmitting a Heartbeat Response Message. If you miss four consecutive Heartbeat Response Messages, you will be disconnected with a Goodbye Message and will have to send another Login Request Message to get back in.

Heartbeat Response Message Format			
Name	*Len*	*Value*	*Notes*
Type	1	"I"	Tells the OUCH machine that this is a Heartbeat Response.

3.3.4 Rewind Request Message

The Rewind Request Message empowers you to recover from message loss. When your OUCH machine receives a Rewind Request Message from you it will immediately begin re-sending messages to you starting with the sequence number you requested. It will keep sending these messages as fast as it can until it is all caught up to the current sequence number. This is nice because your application can treat these re-sent messages just as if they were virgin messages, no additional retransmission logic is necessary.

If you request a sequence number higher than the current sequence number, you will cancel any retransmissions in progress and will not receive any Outbound Sequenced Message until a new one is generated.

You should only request a rewind if you receive a message with a higher sequence number than you expect and quietly ignore messages with lower sequence numbers than expected to avoid getting into an oscillation where consecutive Rewind Requests trigger new ones.

Heartbeat Messages sent during a rewind will indicate the Sequence Number of the last message transmitted as part of the rewind to prevent extraneous rewind requests and make it easier on the client to read in rewinds.

Rewind Request Message Format			
Name	*Len*	*Value*	*Notes*
Type	1	"W"	This tells your OUCH machine that this is a Rewind Request Message.
Sequence	10	numeric	This is the Sequence Number of the next message you wish to receive.

3.4 Outbound Un-Sequenced Messages

During an OUCH connection, the machine can send these messages to you to alert you to changes in the connection status.

3.4.1 Welcome Message

After sending a Login Request Message with the correct Account Name and Password fields, you will be greeted with a Welcome Message telling that your credentials have been accepted and that data will now begin to flow.

Welcome Message Format			
Name	**Len**	**Value**	**Notes**
Type	1	"W"	Tells you that you are in. Your login was accepted and the data will begin to flow.
Version	5	numeric	OUCH Version. This five-digit numeric gives you the three-digit major and two-digit minor revision level of this machine. For example, "100" would represent that this machine supports version 1.00 of the OUCH spec. This is for reference purposes only.
Copyright	60	text	This is a freeform copyright message.

3.4.2 Goodbye Message

A Goodbye Message is sent to let you know that your connection is being terminated. Once you get a Goodbye Message, you will have to send another Login request Message before sending or receiving any more data.

Goodbye Message Format			
Name	Len	Value	Notes
Type	1	"G"	Tells you just got hung up on.
Reason	1	alpha	This lets you know exactly why you are getting this Goodbye Message. See the table of Goodbye Reason Specifiers below to find out why you were hung up on.

Goodbye Reason Specifiers	
Specifier	**Event Explanation**
"O"	Requested. Logout Request Message received.
"E"	Expired. You failed to respond to four consecutive Heartbeat Messages with any Heartbeat Response Messages. Be more thoughtful next time.

"J"	Rejected. You sent a Login request Message, but your Account Name or Password was rejected. Check your spelling and try again.
"N"	Goodnight. It's time to go home. NASDAQ is shutting down for the night. See you bright and early tomorrow.
"M"	Manual. An NASDAQ Supervisor terminal has manually logged you out. This will probably only happen at your own request.

4 Debug Messages

Debug Messages may be sent by the OUCH machine. These messages are intended to help debug problems and are meant for human consumption only. These messages should be ignored or maybe displayed by your software. A debug message always starts with a plus sign, and is followed by a variable number of characters.

You can request to have debug messages sent to your Account Name during software development. Debug messages can give you a human readable reference to keep track of what's going on during your connection. Once in production, you will probably want to turn off debug messages to conserve bandwidth.

Debug Message Format			
Name	*Length*	*Value*	*Notes*
Type	1	"+"	Identifies this as a Debug Message. Please ignore it.
Text	variable	text	This is a freeform English text message.

5 Support

If you have any questions about this specification, please email
tradingservices@nasdaqomx.com.

6 Revision History

6.1 Version 0.3 - 8/4/97

6.1.1 Initial dissemination to a select few developers.

6.2 Version 0.4 - 9/15/95

6.2.1 Miscellaneous spelling and format fixes.

6.2.2 Changed all references to an OUCH port to OUCH machine; there was too much confusion with OUCH ports and TCP/IP port numbers.

6.2.3 Added to Section 4.0 to make it clear that Inbound Messages are not sequenced, they can and should be resent if necessary.

6.2.4 Made the Reject Order and Reject Cancel messages un-sequenced. This makes more sense conceptually especially when dealing with mirrored machines. A rejected order (or cancel) was never accepted by INET; it has no reference number and should not have a sequence number either. If one machine of a mirrored pair enters an order that is rejected, the other machine will get no messages about the rejected order. This makes sense because the rejected order did not affect the state of either machine in any way. It is as if rejected orders never existed from INET's point of view. You can re-send a rejected order as many times as you like and you will keep getting identical reject messages back.

6.2.5 Added Debug Messages to help people see what was going on. These also make it very nice if you Telnet into an OUCH machine because you can see what's going on.

6.2.6 Added a line feed at the end of all outbound messages. If you've already coded to just returns, you can just ignore out any line feeds. I ignore them on incoming messages, so needn't send them either. I know it wastes a little bandwidth, but it's worth it. Now people can use line based functions to read and parse the stream and everything looks so nice and neat when you telnet into an OUCH port. It's all about aesthetic, man.

6.3 Version 0.5 - 9/17/97

6.3.1 Changed the Reason field on the Broken Trade Message. It is now 4 long rather than 8 and the Reason Specifiers have been condensed. I did this to be more consistent with the Reduce Order Reasons. I also reduced the length of the reason field on the Reject Cancel Message.

6.3.2 Added the legend that "malformed numbers produce undefined results" under Architecture. For now, I do a standard atoi() type conversion so trailing spaces are

tolerated, but that may change in the future.

6.3.3 Added a new Reject Order Reason "THRSHOLD" that indicates the number of shares in this order exceeds the preset threshold for this Account.

6.3.4 Made the Firm Field in the Enter Order message mandatory. Before you could leave this filed blank and you would get the default firm for this Account. This way is a double check to make certain that you are entering the right orders in the firm on the right account. If this is a problem, I can override it from here, but your software should know what firm it's entering orders for.

6.3.5 Added Current Restrictions section to detail the limitations of the current OUCH software revision.

6.3.6 Added word "incremental" to the description of the Executed Shares field of the Executed Order message to make it clear that this is the number of shares just executed, not the cumulative number of shares executed.

6.3.7 Removed the paragraph that said that you could send a Rewind Request Message with a sequence number of zero. This didn't add any functionality above requesting a rewind to Sequence Number 1, so I took it out. This way we'll have zero for a special use in the future.

6.3.8 Corrected the word "Login" to "Logout" on Logout Message table legend.

6.4 Version 1.0 - 11/10/97

6.4.1 Changed the behavior of the Cancel Order Message. Before you sent the number of shares to cancel, now you must send the number of shares you want to have remaining after the cancel. Sending zero shares cancels the entire open order. This is much better because you can now send the cancel message multiple times with the same effect, where before each cancel message would have cumulatively canceled more shares.

6.4.2 Corrected the header on the Cancel Order Message table.

6.4.3 Added a few paragraphs clarifying retransmissions in the description of Un-Sequenced Inbound Messages.

6.4.4 Removed the "LOCKED" Order Reject Reason. Locking orders are now passed to the INET execution engine to give them the opportunity to execute against existing orders. If the locking order is not executed, you will get a Canceled Order Message back immediately with a reason of "#LCK".

6.4.5 Added "#LCK" reason to Canceled Order Message. See above.

6.4.6 Added a sentence to the description of the Logout Request Message saying that the OUCH machine will close your TCP/IP connection.

6.4.7 Added a few lines to the description of the Rewind Request Message.

6.5 Version 1.01 - December 2, 1997

6.5.1 Added a line to the Login Message description saying that you only get 30 seconds to log in once you have established a connection.

6.5.2 Corrected the description of the Sequence Number field in the Heart Beat Message to indicate that you get the next *expected* sequence number.

6.5.3 Deleted Known Restriction 9.5. OUCH now handles locking and crossing orders perfectly.

6.5.4 Added a line to the description of the #LCK Order Reject Reason describing the fact that you can now freely lock or cross the primary market outside normal market hours.

6.5.5 NO DOCUMENT CHANGE: OUCH will now actually pass the #LCK Order Reject Reason when appropriate. While this Reason was in the spec previously, I filtered it out so it never actually appeared in the protocol.

6.5.6 NO DOCUMENT CHANGE: OUCH now logs the IP address and socket number of the connection source in it's log file.

6.6 Version 1.02 - January 26, 1998

6.6.1 Changed the "Displayed" field in the Enter Order Message to "Display" and added the value "S" for new subscriber only orders.

6.6.2 Changed the "Displayed" field in the Order Accept Message to "Display" and added the value "S" for new subscriber only orders.

6.6.3 Added a description line for Subscriber Only orders in the section describing Minimums & Hidden Orders.

6.6.4 Modified the description of the "DISPLAY" reject reason in the Order Reject Message to account for the new "S" value. The "S" indicates a subscriber only order.

6.6.5 Removed the restriction on placing hidden orders from the Current Restrictions list. You can now use hidden orders.

6.7 Version 1.03 – November 23, 1998

6.7.1 Changed the name of the field in the Enter Order Message and Accept Order Message from "Principal/Agency" to "Capacity".

6.7.2 Added a new value to the Capacity field (previously the "Principal/Agency" field) in the Enter Order Message and Accept Order Message. The new value "R" lets firms enter trades in a riskless capacity for trade reporting.

6.7.3 Changed Immediate Or Cancel orders that would lock/cross to correctly return a reason of "#IOC" if they do not execute. Before they returned the reason "#LCK" even though they were Immediate or Cancel orders so they could not have locked the market.

6.7.4 Miscellaneous spelling and wording changes.

6.8 Version 1.04 – December 12, 1999

Version 1.04 is a minor release necessitated to support listed and after hours trading on INET. Since these changes only affect reason code specifiers they should not require

any code changes to client software. Furthermore, you'll never even see these changes unless you try and enter an order in a listed security or the extended trading session.

6.8.1 Added "DOWNTICK" as a reason specifier to the Rejected Order Message to allow INET to accept short sales in listed securities in compliance to exchange short sale rules.

6.8.2 Added "#DNT" as a reason specifier to the Cancel Order Message to allow INET to cancel open short sell orders in listed securities when required to comply with exchange short sale rules.

6.8.3 Added "NOQSR" as a reason specifier to the Rejected Order Message to allow INET to reject orders entered after by firms that clear using ACT at a time when NASDAQ's ACT System is not available.

6.8.4 Added "MASTER" as a reason specifier to the Rejected Order Message to allow INET to reject orders if the INET's internal security master system is not available. Hopefully subscribers will never see this reason specifier.

6.8.5 Changed the incorrect behavior for orders entered with a time in force of 99998 after the close of the primary market. In previous versions, orders entered with a time in force of 99998 after the close of the primary market were incorrectly accepted as regular day (99999) orders. These orders will be accepted as an IOC.

6.9 Version 1.05

6.9.1 Added support for the INET Trade Reporting Service Bureau. This allows subscribers to use INET's ACT trade reporting infrastructure to report listed and NASDAQ trades to the tape.

6.9.2 Removed the implementation notes and a few spelling changes. Updated all email addresses to Island.com rather than isld.com and removed all personal references.

6.9.3 Added Match Number to the end of the Execution and Break messages. This finally makes it possible to implement breaks correctly. It's also nice to know the Match Number for a trade because it can be cross-referenced to ITCH and to the clearing contract sheets.

6.9.4 Removed Text Messages since INET currently uses out-of-band email lists for this function.

6.9.5 Removed Development Guidelines section because it was not particularly useful.
6.9.6 Added qualifications to the Alpha-num data type specifying that it will only accept digits and letter and that letters are case sensitive.
6.9.7 Removed the "On minimums" section because it did not belong in this spec.
6.9.8 Added the "Minimum Increment" field into the Enter Order and Order Accepted messages to let people control the decimal/fraction rounding on an order-by-order basis. This is particularly handy for service bureaus are not.

6.10 Version 1.05 Revised

6.10.1 Replaced all references to Island with INET

6.10.2 Added new codes for Short Sale with Affirmative Determination and Short Sale without Affirmative Determination

6.11 Version 1.05 Revised

6.11.1 Due to change in handling HOLDRS, section 2.1.2.1, *Accepted Order Message, Shares* now returns *Shares Accepted,* not *Shares Entered*

6.12 Version 1.05 Revised January 10, 2007

6.12.1 Display Field: Updated values and added value "C"
6.12.2 Updated Liquidity Flags. Added Liquidity Flags Table
6.12.3 "Minimum Increment" changed to "Reserved for NASDAQ". Value will be blank
6.12.4 "Executing Firm (FYI)" changed to "Reserved for NASDAQ". Value will be blank
6.12.5 Changed references to "INET" to "NASDAQ"

6.13 Version 1.05 Revised October 5th, 2007

6.13.1 Updated order message buy/sell indicator where "E" specifies sell short, client affirms ability to borrow securities in good deliverable form for delivery within three business days

6.14 Version 1.06 Revised March 14, 2008

6.14.1 Updated values for Display field
6.14.2 Updated email address for support-related issues
6.14.3 Updated values for Display field. Added value "Z" to Accepted Order Message and removed value "S"
6.14.4 Updated Liquidity Flags for value "J"

6.15   Version 1.06 revised – October 21st, 2008

6.15.4 Removed "A" value from display field for order entry

Formatted: Bullets and Numbering

Formatted: Font: Not Bold