



QIX for OTCBB
Programming Specification

September 26, 2005

2005-01

Confidentiality/Disclaimer

This **DRAFT** Specification is being forwarded to you strictly for informational purposes solely for the purpose of developing or operating systems for your use that interact with systems of The NASDAQ Stock Market, Inc. (NASDAQ) and its affiliates (collectively, the Corporations). This specification is proprietary to NASDAQ. NASDAQ reserves the right to withdraw, modify, or replace the specification at any time, without notice. No obligation is made by NASDAQ regarding the level, scope, or timing of NASDAQ's implementation of the functions or features discussed in this specification. The specification is "AS IS," "WITH ALL FAULTS" and NASDAQ makes no warranties, and disclaims all warranties, express, implied, or statutory related to the specifications. THE CORPORATIONS ARE NOT LIABLE FOR ANY INCOMPLETENESS OR INACCURACIES. THE CORPORATIONS ARE NOT LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES RELATING TO THE SPECIFICATIONS OR THEIR USE. It is further agreed by you by using this specification, that you agree not to copy, reproduce, or permit access to the information contained in, the specification except to those with a need-to-know for the purpose noted above. Copyright 2005, The NASDAQ Stock Market, Inc., as an unpublished work. All Rights Reserved.

Contents

1 Overview	4
1.1 Publication History	5
1.2 NASDAQ Customer Support Information.....	5
1.3 Data Standards.....	6
1.3.1 Alphabetic, Numeric, and Alphanumeric Fields	6
1.3.2 Price Fields	6
1.3.3 Symbology	6
1.3.4 Server Tracking Number Format	6
1.4 Message Sequencing	7
2 Network Configuration	8
2.1 Configuration Requirements	8
2.2 Example Configurations	9
2.2.1 ECN Configuration with a Primary and Secondary Site.....	9
2.2.2 ECN Alpha Split Configuration with a Primary and Secondary Site	10
2.2.3 Market Maker/ECN Configuration with a Primary and Secondary Site.....	11
3 Session Protocol	12
3.1 Session Connection	12
3.2 QIX Logical Packets	13
3.3 Session ID Format	14
3.4 Session Messages	15
3.4.1 Login Request	15
3.4.2 Login Accepted.....	17
3.4.3 Login Rejected	18
3.4.4 Logout Request	19
4 Application Protocol	20
4.1 OTCBB Query/Response Message Formats	20
4.1.1 Market Maker Registration 4298 Version 00.....	20
4.1.2 Withdraw Quote 4315 Version 00.....	24
4.1.3 Quote Update 4485 Version 00.....	26
4.2 Debug Packet	31
4.3 Echo Packet	31
4.4 Heartbeat Packets.....	32
4.4.1 QIX Server	32
4.4.2 QIX Client	32

1 Overview

This document describes QIX, a NASDAQ protocol that provides a fast, reliable point-to-point connection from NASDAQ Market Center to your site. This document contains the QIX session level protocols used by NASDAQ, the application functionality currently supported by QIX, and the requirements for using the QIX protocol. This document does not consider messages transmitted via other entry points into NASDAQ.

This release of QIX supports:

OTCBB Query/Response Messages

- Market Maker Registration
- Quote Update
- Withdraw Quote

As new QIX functionality becomes available, you will receive technical updates detailing that functionality. At the same time the technical update is distributed, this programming specification will be updated on the NASDAQ Trader web site.

1.1 Publication History

Date	Version	Enhancements
September 26, 2005	2005-1	1. Formatting updated, published
August 8 th , 2005	2005-1	2. Initial version of OTCBB specific document
July 23, 2004	draft	Initial publication of this programming specification.

1.2 NASDAQ Customer Support Information



NASDAQ QIX Support

800-243-4284



NASDAQ Subscriber Services

800-777-5606



NASDAQ Trading Services

800-219-4861

1.3 Data Standards

At the session and application levels, all data fields will be fixed length.

1.3.1 Alphabetic, Numeric, and Alphanumeric Fields

All alphabetic, numeric, and alphanumeric fields are left-justified and right-padded with spaces, except where noted.

1.3.2 Price Fields

1.3.2.1 NASDAQ Market Center Trading and OTCBB

All price fields are 10 characters in length with the first 6 characters reserved for the dollar portion and the last 4 reserved for cents. The decimal point is implied. Dollar amounts should be prepended with zeroes. Cent amounts should be appended with zeroes. For example, \$12.34 is 0000123400.

1.3.3 Symbology

Listed security symbols are entered in CMS format, which uses space delimiters between the root symbol and any suffix. For example, OXY PRB.

1.3.4 Server Tracking Number Format

The server provides the Server Tracking Number to you on a UM connection in all sequenced messages. The Sequence Number portion of the Server Tracking Number is used in the Login Request Message for requesting rewind service.

Name	Offset	Length	Value	Description
Sequence Number	0	8	numeric	Right-justified, zero-filled.
Reserved for NASDAQ	8	2	alphanumeric	
Reserved for NASDAQ	10	2	alphanumeric	

1.4 Message Sequencing

With the release of version 2.0, NASDAQ will support optional inbound message sequencing on the Q/R connection. You can use this feature on a message-by-message basis; that is, you can choose to designate some inbound packets as sequenced and others as unsequenced.

Each query message contains a new Packet Type value, *s*, for sequenced and a new field, Client Sequence Number. You supply the incoming sequence number in messages that you designate as sequenced.

QIX monitors the sequence number for message gaps. If you send a transaction with a sequence number greater than expected, QIX will send a new Message Gap Detected Reject message. QIX will continue to discard all messages until the expected sequence number is received or until you reset the sequence number via the new Reset Sequence Number message. When QIX receives a message with a sequence number less than expected, a debug packet will be returned with the Text field = "Expected Sequence nnnnnnnn, Received Sequence nnnnnnnn". The client session is then terminated.

The Client Sequence Number will be reset to 1 by QIX each morning.

Example #1

1. You connect and send a Login Request to QIX on the Q/R connection.
2. QIX validates the Login and responds with a Login Accepted message containing the expected sequence number, in this case, 1.
3. You then transmit sequence numbers 1 and 2 followed by 4.
4. QIX detects the gap and responds with a Message Gap Detected Reject.
5. If messages 5 and 6 are in flight, they will be discarded with no response back to you.
6. You send in sequence number 3 and continue.
or
6. You send a Reset Sequence Number message to reset the number.

Example #2

1. You connect and send a Login Request to QIX on the Q/R connection.
2. QIX validates the Login and responds with a Login Accepted message containing the expected sequence number, in this case, 20.
3. You send a message with sequence number 18.
4. QIX sends a Debug packet with a canned text string and disconnects.

2 Network Configuration

2.1 Configuration Requirements

All participants in QIX will be required to establish a minimum of two separate TCP/IP connections, each configured to a different port number.

A UM delivery connection is used to transmit unsolicited messages to you. This connection also supports session level and control level messages, which are unsequenced.

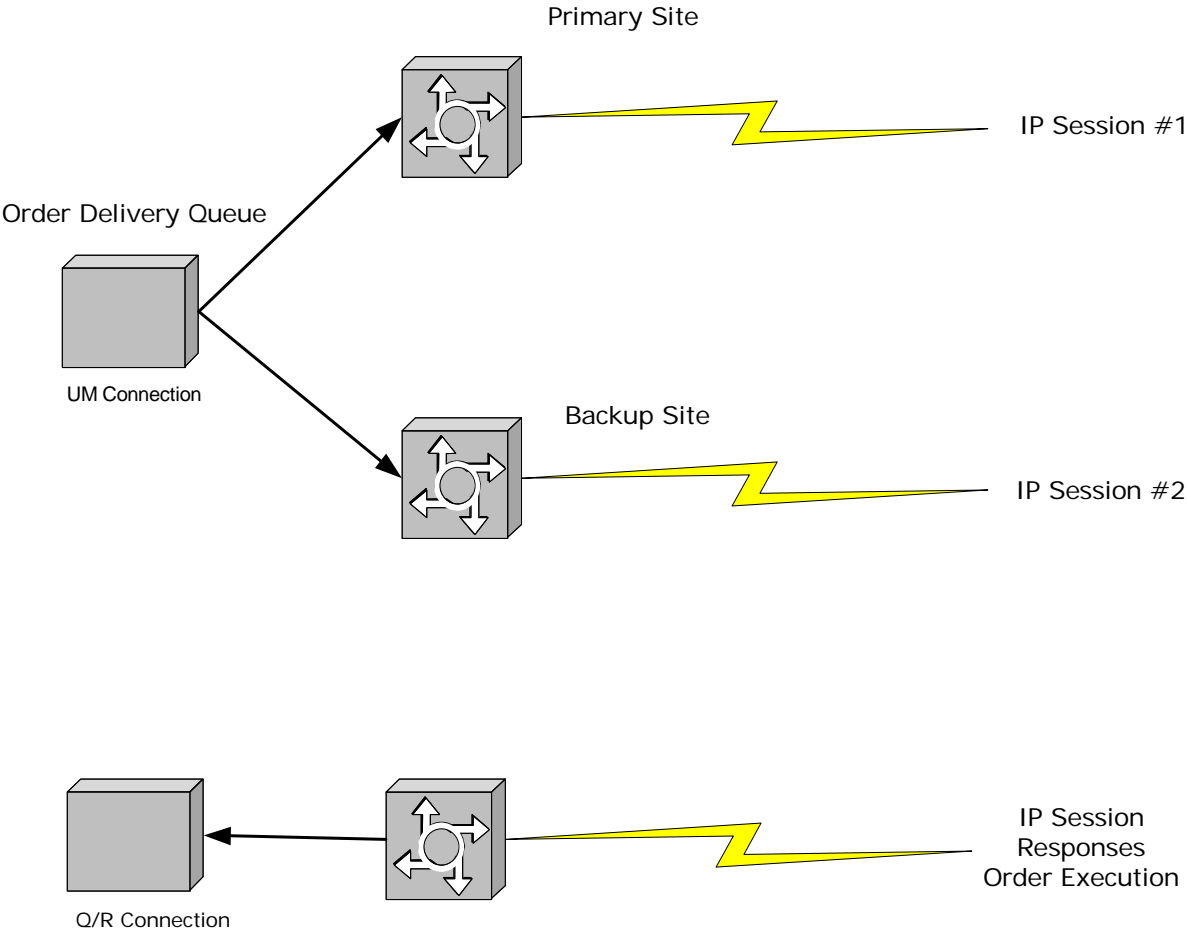
In order to receive your proprietary quote updates, you must establish a separate UM connection. Only Proprietary Quote UM traffic is sent on this connection. These messages are designated as sequenced, but sequence numbers are not applicable to this connection. When you log on to the Proprietary Quote connection, the server will begin transmitting the last value cache figures for each security that you currently hold a proprietary position in. This is true of your initial login as well as subsequent logins intraday. This connection also supports session level and control level messages, which are unsequenced.

You use the Q/R connection to send traffic to and receive responses from NASDAQ applications. The Query/Response traffic to NASDAQ applications can be sequenced or unsequenced inbound. No rewind capability is offered on a Q/R connection. This connection also supports session level messages and control level messages that are unsequenced.

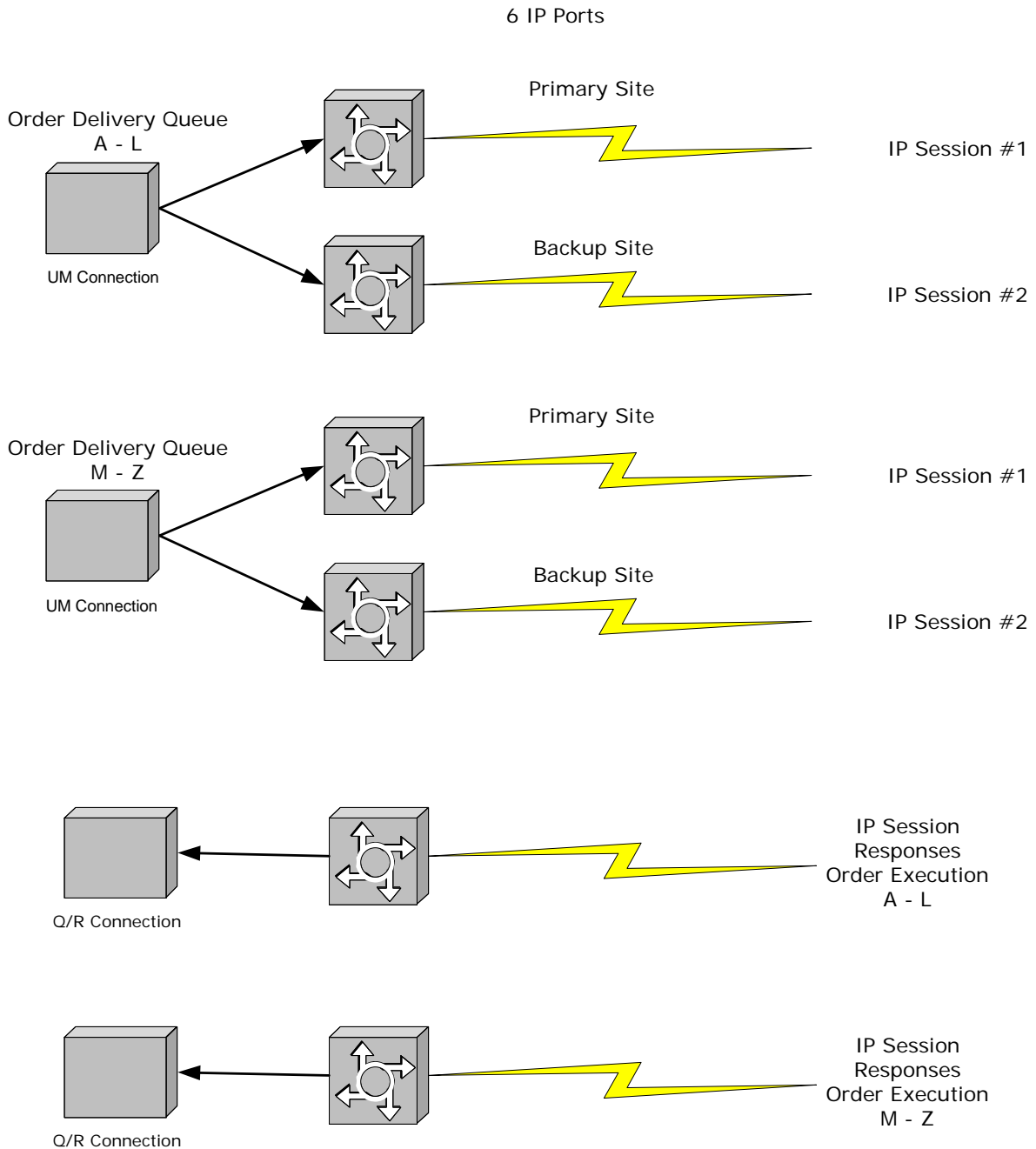
2.2 Example Configurations

2.2.1 ECN Configuration with a Primary and Secondary Site

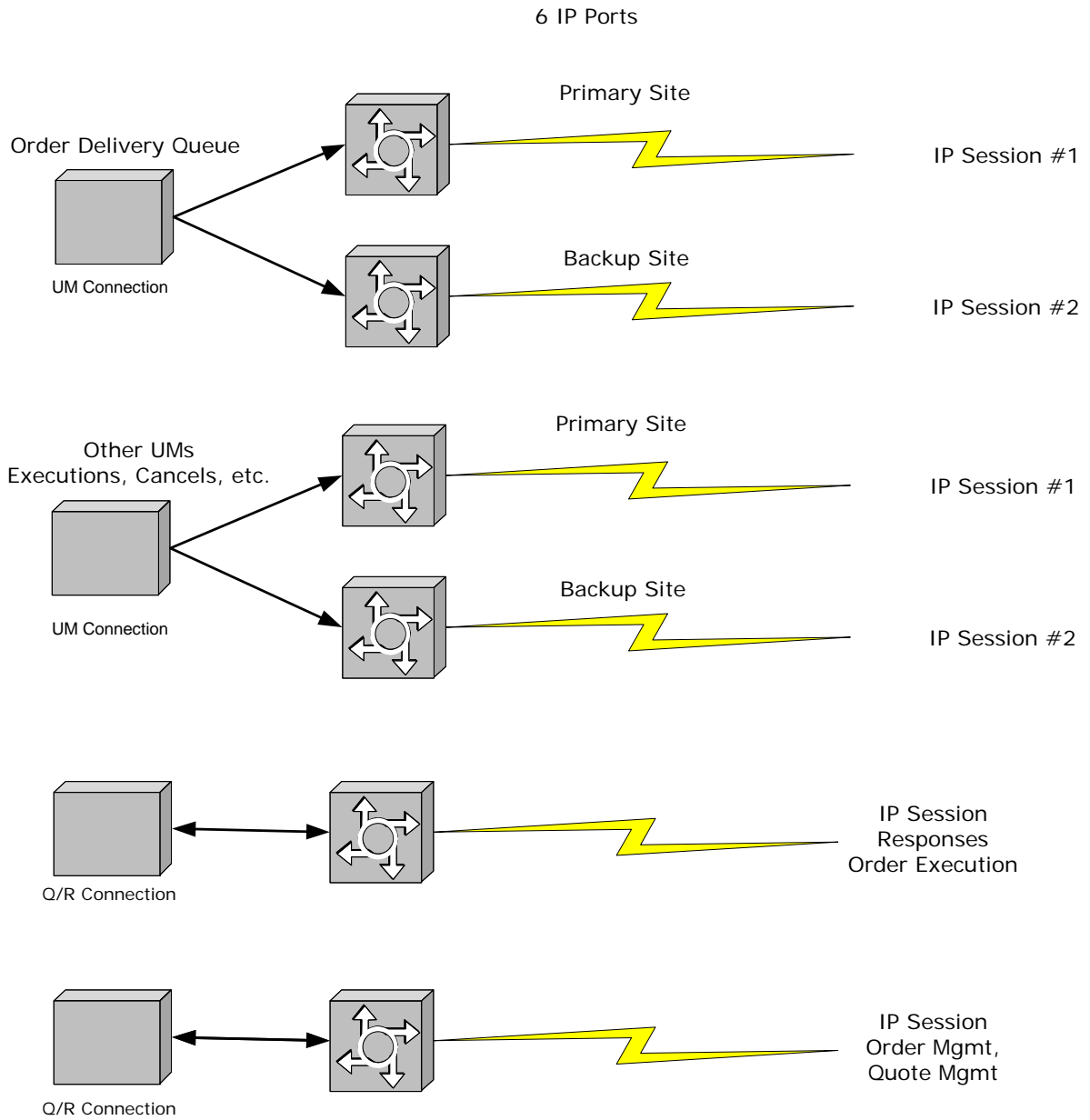
3 IP Ports



2.2.2 ECN Alpha Split Configuration with a Primary and Secondary Site



2.2.3 Market Maker/ECN Configuration with a Primary and Secondary Site

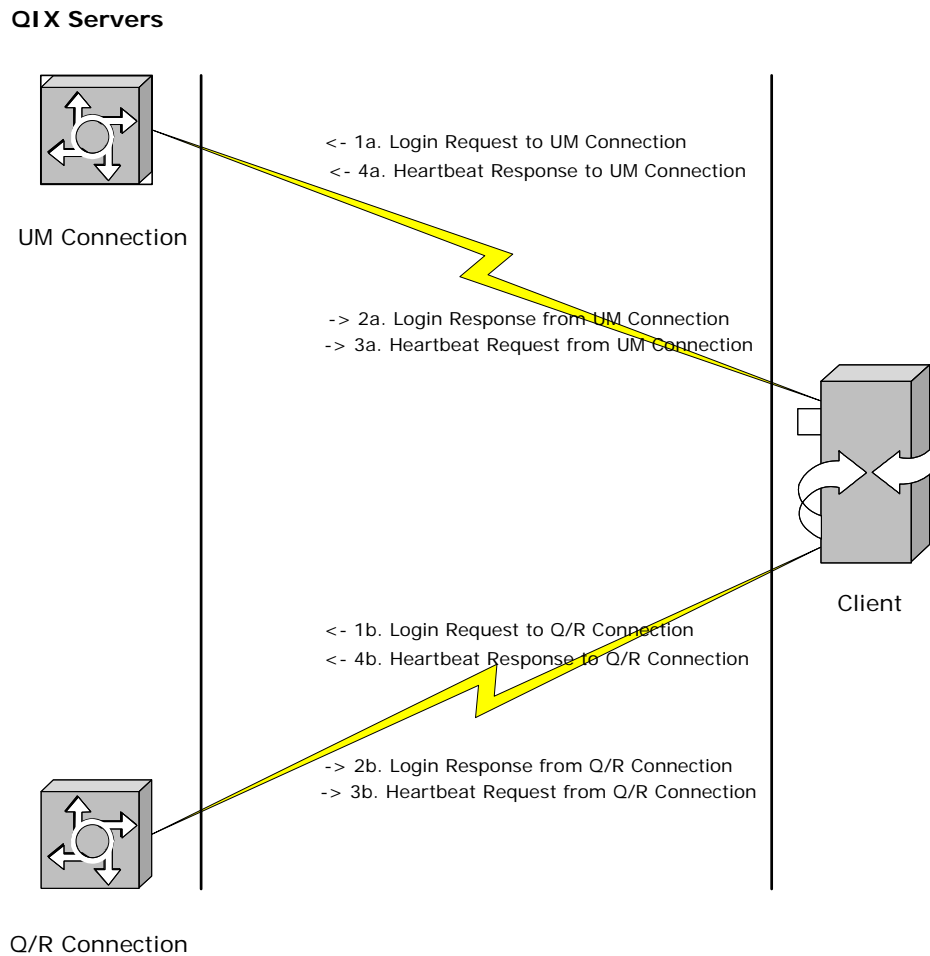


3 Session Protocol

QIX uses a new, lightweight point-to-point session level protocol that is described in this section.

3.1 Session Connection

1. Immediately upon establishing a new TCP/IP socket connection to the server, you must send a Login Request Message to the UM Connection (a) and the Query/Response (Q/R) Connection (b).
2. You will receive a Login Response from the UM Connection (a) and the Q/R Connection (b).
3. You will receive Server Heartbeat Messages every five seconds from the UM Connection (a) and the Q/R Connection (b).
4. You respond immediately with your Heartbeat Message to the UM Connection (a) and the Q/R Connection (b).



3.2 QIX Logical Packets

The QIX client and server communicate by exchanging a series of logical packets.

Each QIX logical packet begins with a 4-byte, ASCII, right-justified, and zero-filled **message length** (includes the trailer, but does not include itself) followed by a single-byte **packet type**, the **transaction code**, the **message version number**, the **server tracking number**, and 4 bytes of **Reserved data** for UMs or the **client tracking number** and **client sequence number** for Q/R.

A variable length **payload** follows. The payload may not contain the linefeed character (ASCII 10 decimal, 0x0A hex) or the pipe character (ASCII 124 decimal, 0x7C hex).

The final component is a **standard trailer** consisting of a single terminating linefeed character (ASCII 10 decimal, 0x0A hex).

3.3 Session ID Format

The Session ID is defined as a 10-character alphanumeric string with this format:

Name	Offset	Length	Value	Description
Firm ID	0	4	alphanumeric	Firm Name
Date	4	3	DDD	Julian day
Session Type	7	1	alphabetic	Session Type indicator. s = the session is recoverable (the UM connection) u = the session is non-recoverable (the Q/R connection)
Session Number	8	2	alphanumeric	System-assigned session number.

This Session ID will be unique. It will be derived from information in the Firm ID and Session Number in the user configuration file. It is returned to you in the Login Accepted message.

3.4 Session Messages

3.4.1 Login Request

You must send a Login Request Message immediately upon establishing any TCP/IP socket connection to the server. The session will be terminated within two heartbeat intervals if the Login Request Message is not received. If the server receives any other type message (other than a Debug Packet) any time a Login Request Message is expected, the connection will be dropped. Note: the reception of the Debug Packet does not satisfy the requirement to receive a Login Request Message within two heartbeat intervals. This message type is valid on all connections.

Sample UM Connection Login Request

```
0026IHERO    1234567800000000<\n>
```

Message Length	Packet Type	Username	Password	Sequence Number	Standard Trailer
0026	I	HERO����	12345678	00000000	<\n>

Sample QR Connection Login Request

```
0026IHERO    12345678            <\n>
```

Message Length	Packet Type	Username	Password	Sequence Number	Standard Trailer
0026	I	HERO����	12345678	��������	<\n>

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	I (lower case L) = Login Request Packet
Username	5	8	
Password	13	8	

Field Name	Offset	Length	Value / Comments
Sequence Number	21	8	<p>This field pertains to UM connections only and is ignored on Q/R connections.</p> <p>For a standard UM connection, this field can contain either a valid sequence number (right-justified, zero-filled) or spaces.</p> <p>Spaces indicate that you want to start receiving data where it left off, that is, the next message in sequence.</p> <p>If a valid sequence number is supplied, the server begins to transmit data, starting with the next message following the number designated in the sequence number.</p> <p>If a sequence number of all zeroes is supplied, the server will rewind to start of day and begin transmitting up to the current message.</p> <p>For a proprietary quote UM connection, this field is not applicable at this time. The server will begin transmitting the last value cache figures for each security that you currently hold a proprietary position in.</p>
<i>Standard Trailer</i>	29	1	

3.4.2 Login Accepted

The QIX server sends a Login Accepted message in response to a valid Login Request from you. This message type is valid on all connections.

Sample UM Login Accepted Response

```
0020aHERO355sK000000001<\n>
```

Message Length	Packet Type	Session ID	Sequence Number	Standard Trailer
0020	a	HERO355sK0	00000001	<\n>

Sample QR Login Accepted Response

```
0020aHERO355uK000000001<\n>
```

Message Length	Packet Type	Session ID	Sequence Number	Standard Trailer
0020	a	HERO355sK0	00000001	<\n>

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	a = Login Accepted Packet
Session ID	5	10	The ID of the session that is now logged into.
Sequence Number	15	8	For sequenced UM sessions, specifies the next sequence number to be transmitted by the server. For sequenced Q/R sessions, specifies the next expected Client Sequence Number on this connection.
<i>Standard Trailer</i>	23	1	

3.4.3 Login Rejected

The QIX server sends a Login Rejected message in response to an invalid Login Request from you. This message type is valid on all connections.

Sample Login Rejected Response

0004j01<\n>

Message Length	Packet Type	Reject Reason Code	Standard Trailer
0004	j	01	<\n>

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	j = Login Rejected Packet
Reject Reason Code	5	2	01 = Not Authorized. There was an invalid username and password combination in the Login Request message. 02 = Session not available. 03 = Invalid Sequence Number. 04 = Message Not Available. The Sequence Number supplied in the Login Request is valid but the message is not available.
<i>Standard Trailer</i>	7	1	

3.4.4 Logout Request

You send a Logout Request message to request that the connection be terminated. The server will immediately terminate the connection and close the associated TCP/IP socket. This message type is valid on all connections.

Sample Logout Request

0002o<\n>

Message Length	Packet Type	Standard Trailer
0002	o	<\n>

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	o = Logout Request Packet
<i>Standard Trailer</i>	5	1	

4 Application Protocol

The application level messages implement a new API format based on the current NASDAQ message definitions.

4.1 OTCBB Query/Response Message Formats

OTCBB query/response messages can be sequenced or unsequenced inbound and are valid on the query/response connection only.

4.1.1 Market Maker Registration 4298 Version 00

This message allows a market maker firm to register as a market maker for a specified OTCBB security.

Sample OTCBB Market Maker Registration Request

0044u429800121609000017**XXXXXXXXXXXX**MSEV**XXXXXXXXXXXX**<\n>

Message Length	Packet Type	Transaction Code	Version Number	Client Tracking Number	Client Sequence Number	Security Symbol	Market Maker Location	Filler	Standard Trailer
0044	u	4298	00	121609000017	XXXXXXXXXXXX	MSEV XXXXXXXXXXXX	X	X	<\n>

4.1.1.1 Request

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	s = sequenced u = unsequenced data packet
Transaction Code	5	4	4298
Version Number	9	2	00
Client Tracking Number	11	12	Alphanumeric, required to be unique, customer assigned. Echoed back in the 4298 response.
Client Sequence Number	23	8	If packet type = s, next message sequence number on this connection. Else, space-filled.
Security Symbol	31	14	Required.
Market Maker Location	45	1	Valid values are A-Z, &, a (arbitrage), f (international), p (preferred), v

Field Name	Offset	Length	Value / Comments
			(convertible), or space.
Filler	46	1	space
Standard Trailer	47	1	

Sample OTCBB Market Maker Registration Response

0066u4298001216090000170600018300020041216142656670000200412160930000b<\n>

Message Length	Packet Type	Transaction Code	Version Number	Client Tracking Number	Result code	Close Time	Last Update Timestamp	Effective Date
0066	u	4298	00	121609000017	06000	183000	20041216142656670000	20041216

Open Time	Filler	Standard Trailer
093000	b	<\n>

4.1.1.2 Response

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	u = unsequenced data packet
Transaction Code	5	4	4298
Version Number	9	2	00
Client Tracking Number	11	12	Alphanumeric, from the request.
Result code	23	5	0x000 x = source code 6 = normal host accept 7 = conditional accept – session level pacing warning 8 = conditional accept – session level pacing implemented
Close Time	28	6	Closing Time in HHMMSS format or all spaces.
Last Update Timestamp	34	20	YYYYMMDDHHMMSSNNNNNN.
Effective Date	54	8	YYYYMMDD. Defaults to current date.

Field Name	Offset	Length	Value / Comments
Open Time	62	6	Opening Time in HHMMSS format or all spaces.
Filler	68	1	space
<i>Standard Trailer</i>	69	1	

4.1.2 Withdraw Quote 4315 Version 00

This message allows a market maker firm to withdraw its position from a specified OTCBB security.

Sample OTCBB Withdraw Quote Request

0042u431500121609000017**XXXXXXXXXX**MSEV**XXXXXXXXXX**<\n>

Message Length	Packet Type	Transaction Code	Version Number	Client Tracking Number	Client Sequence Number	Security Symbol	Standard Trailer
0042	u	4315	00	121609000017	XXXXXXXXXX	MSEV XXXXXXXXXX	<\n>

4.1.2.1 Request

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right justified, zero filled.
Packet Type	4	1	s = sequenced u = unsequenced data packet
Transaction Code	5	4	4315
Version Number	9	2	00
Client Tracking Number	11	12	Alphanumeric, required to be unique, customer assigned. Echoed back in the 4315 response.
Client Sequence Number	23	8	If packet type = s, next message sequence number on this connection. Else, space-filled.
Security Symbol	31	14	Required.
Standard Trailer	45	1	

Sample OTCBB Withdraw Quote Response

0046u4315001216090000170600020041216142939270000¸<\n>

Message Length	Packet Type	Transaction Code	Version Number	Client Tracking Number	Result code	Last Update Timestamp	Filler	Standard Trailer
0046	u	4315	00	121609000017	06000	20041216142939270000	¸	<\n>

4.1.2.2 Response

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right justified, zero filled.
Packet Type	4	1	u = unsequenced data packet
Tran Code	5	4	4315
Version #	9	2	00
Client Tracking #	11	12	Alphanumeric – from the request
Result code	23	5	0x000 x = source code 6 = normal host accept 7 = conditional accept – session level pacing warning 8 = conditional accept – session level pacing implemented
Last Update Time Stamp	28	20	Format is YYYYMMDDHHMMSSNNNNNN.
Filler	48	1	Space
<i>Standard Trailer</i>	49	1	

4.1.3 Quote Update 4485 Version 00

This message allows a market maker firm to establish and update quotes for a specified OTCBB security. The transaction also allows a position to be closed and re-opened.

Sample OTCBB Quote Update Request

0078u448500121609000018MSSEV0000000300000000500NNNNN<\n>

Message Length	Packet Type	Transaction Code	Version Number	Client Tracking Number	Client Sequence Number	Security Symbol	Open Close Code	Bid Direction	Bid Quote Price	Bid Size Quantity
0078	u	4485	00	121609000018	MSSEV	0000000300	0	0	0000000300	NNNNN

Ask Direction	Ask Quote Price	Ask Size Quantity	OTCBB Bid Wanted Flag	OTCBB Offer Wanted Flag	OTCBB No Quote Bid Flag	OTCBB No Quote Ask Flag	OTCBB Unsolicited Flag	Standard Trailer
0	0000000500	NNNN	N	N	N	N	N	<\n>

4.1.3.1 Request

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	s = sequenced u = unsequenced data packet
Transaction Code	5	4	4485
Version Number	9	2	00
Client Tracking Number	11	12	Alphanumeric, required to be unique, customer assigned. Echoed back in the 4485 response.
Client Sequence Number	23	8	If packet type = s, next message sequence number on this connection. Else, space-filled.
Security Symbol	31	14	Required.

Field Name	Offset	Length	Value / Comments
Open Close Code	45	1	Used to open or close a position/quote. O = Open C = Close space = not Open/Close function
Bid Direction	46	1	Indicates how to treat the Bid Quote Price field. 0 = decrement current bid quote 1 = increment current bid quote space = replace current bid quote
Bid Quote Price	47	10	If Bid Direction field has a value, then the price is treated as an increment/decrement to the current bid quote. Otherwise, the price is used as a replacement Bid Quote Price. all spaces – no price entered
Bid Size Quantity	57	4	Number of round lots in this security associated with the MP Bid Price. all spaces – no size entered
Ask Direction	61	1	Indicates how to treat the Ask Quote Price field. 0 = decrement current ask quote 1 = increment current ask quote space – replace current ask quote
Ask Quote Price	62	10	If Ask Direction field has a value, then the price is treated as an increment/decrement to the current ask quote. Otherwise, the price is used as a replacement Ask Quote Price. all spaces – no price entered
Ask Size Quantity	72	4	Number of round lots in this security associated with the MP Ask Price. all spaces = no size entered
OTCBB Bid Wanted Flag	76	1	N = No action Y = Ask wanted
OTCBB Offer Wanted Flag	77	1	N = No action Y = Bid wanted
OTCBB No Quote Bid Flag	78	1	N = No action Y = Remove bid quote
OTCBB No Quote Ask Flag	79	1	N = No action Y = Remove ask quote

Field Name	Offset	Length	Value / Comments
OTCBB Unsolicited Flag	80	1	N = Not unsolicited Y = Unsolicited
Standard Trailer	81	1	

Sample OTCBB Quote Update Response

0094u44850012160900001806000YMSEVbbb bbbbbb000000030050bb000000050050bb
NNNNN20041216142909380000<\n>

Message Length	Packet Type	Transaction Code	Version Number	Client Tracking Number	Result code	Lock Cross Flag	Security Symbol	Open Close Code	Bid Quote Price	Bid Size Quantity
0094	u	4485	00	121609000018	06000	Y	MSEVbbb bbbbbb	b	0000000300	50bb

Ask Quote Price	Ask Size Quantity	OTCBB Bid Wanted Flag	OTCBB Offer Wanted Flag	OTCBB No Quote Bid Flag	OTCBB No Quote Ask Flag	OTCBB Unsolicited Flag	Accepted Time Stamp	Standard Trailer
0000000500	50bb	N	N	N	N	N	20041216142909380000	<\n>

4.1.3.2 Response

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	u = unsequenced data packet
Transaction Code	5	4	4485
Version Number	9	2	00
Client Tracking Number	11	12	Alphanumeric, from the request.
Result code	23	5	0x000 x = source code 6 = normal host accept 7 = conditional accept – session level pacing warning. 8 = conditional accept – session level pacing implemented
Lock Cross Flag	28	1	N = No lock/cross Y = Quote lock/cross
Security Symbol	29	14	

Field Name	Offset	Length	Value / Comments
Open Close Code	43	1	O = Open C = Close space = not Open/Close function
Bid Quote Price	44	10	The bid price resulting from the quote update.
Bid Size Quantity	54	4	Number of round lots in this security associated with the MP Bid Price.
Ask Quote Price	58	10	The ask price resulting from the quote update.
Ask Size Quantity	68	4	Number of round lots in this security associated with the MP Ask Price.
OTCBB Bid Wanted Flag	72	1	N = No action Y = Ask wanted
OTCBB Offer Wanted Flag	73	1	N = No action Y = Bid wanted
OTCBB No Quote Bid Flag	74	1	N = No action Y = Remove bid quote
OTCBB No Quote Ask Flag	75	1	N = No action Y = Remove ask quote
OTCBB Unsolicited Flag	76	1	N = Not unsolicited Y = Unsolicited
Accepted Time Stamp	77	20	Date/Time of Quote Update. YYYYMMDDHHMMSSNNNNNN.
<i>Standard Trailer</i>	97	1	

4.2 Debug Packet

Either side of a QIX connection at anytime can send a Debug Packet. They are intended to provide human readable text that may aid in debugging problems. Inbound Debug Packets to the server will be logged and ignored. No response is sent to you.

QIX will utilize the Debug Packet to alert you of a problem such as a garbled message or unknown message type. The server will respond with an appropriate error message in the Text field prior to disconnecting the socket.

The maximum message length for a Debug Packet (Packet Type, Text, and Trailer) is 240 bytes.

This message type is valid on both connections.

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	+ = debug packet
Text	5	Variable	Free form human readable text.
<i>Standard Trailer</i>	Text Length + 1	1	

4.3 Echo Packet

Anytime after login, either side of a QIX connection can send an Echo Packet. They are intended to confirm that the peer connection is functional. All that is required of a peer upon receipt of the "e" packet is to change the packet type "e" to packet type "E" before echoing the otherwise unchanged packet in its entirety back to the originating peer.

The maximum message length for an Echo Packet (Packet Type, Text, and Trailer) is 240 bytes.

This message type is valid on both connections.

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	e or E = echo packet
Text	5	Variable	Free form human readable text.
<i>Standard Trailer</i>	Text Length + 1	1	

4.4 Heartbeat Packets

The QIX Server Heartbeat Packet contains a timestamp to be used for performance measurement. The server will send the QIX Server Heartbeat Packet to you every five seconds. You are required to echo back the QIX Client Heartbeat Packet. You never initiate the transmission of the QIX Client Heartbeat packet. It is sent only as a result of receiving the QIX Server Heartbeat Packet. This message type is valid on both connections.

4.4.1 QIX Server

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	h
Timestamp	5	32	
<i>Standard Trailer</i>	37	1	

4.4.2 QIX Client

Field Name	Offset	Length	Value / Comments
Message Length	0	4	ASCII, right-justified, zero-filled.
Packet Type	4	1	r
Timestamp	5	32	Must be an echo of the "h" packet timestamp field.
<i>Standard Trailer</i>	37	1	

The server will send the QIX Server Heartbeat Packet to you every five seconds, whether data is queued or not. You can assume that the link is lost if it does not receive anything for 15 seconds.