tradelogiq

Tradelogiq Markets Inc.
tradelogiq.com

TRADELOGIQ MARKETS INC.

Omega ATS and Lynx ATS
Soup TCP Binary Protocol Specification

v. 1.03
For ITCH 5.0 (Market Data)

Revision History

Date	Revision	Description of Change
October 19, 2017	1.00	Document Created
October 20, 2017	1.01	Edits made to document.
October 24, 2017	1.02	Edits made to document.
Jan 8, 2019	1.03	Updated the **Value** column for the following sections: 1.5.1 Login Accepted Packet Sequence Number 1.6.1 Login Request Packet Requested Sequence Number

Contents

1. Introduction

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

Clients can also send messages to the server via SoupTCPBinary; however, these messages are not sequenced and may be lost in the case of a TCP/IP socket failure.

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

SoupTCPBinary is designed to be used in conjunction with higher lever protocols that specify the contents of the messages that are being delivered. This protocol layer is opaque to the higher-level messages.

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

1.1. SoupTCPBinary Logical Packets

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

Each SoupTCPBinary logical packet has:

- a two byte big-endian length that indicates the rest of the packet
- a single byte header which indicates the packet type
- a variable length payload

Two Byte Packet Length	Packet Type	Variable-length payload

SoupTCPBinary Logical Packet Structure

NOTES:

- The SoupTCPBinary logical packets do not necessarily map directly to physical packets on the underlying network socket; they may be broken apart or aggregated by the TCP/IP stack.
- The SoupTCPBinary protocol does not define a maximum payload length.

1.2. Protocol Flow

A SoupTCPBinary connection begins with the client opening a TCP/IP socket to the server and sending a Login Request Packet. If the login request is valid, the server responds with a Login Accepted Packet and begins sending Sequenced Data Packets. The connection continues until the TCP/IP socket is broken.

Each Sequenced Data Packet carries a single higher-level protocol message.

Sequenced Data Packets do not contain an explicit sequence number; instead both client and server compute the sequence number locally by counting messages as they go.

The sequence number of the first sequenced message in each session is always 1.

When initially logging into a server the client should set the Requested Sequence Number field to 1 and leave the Requested Session field blank in the Login Request Packet. The client will then inspect the Login Accepted Packet to determine the currently active session. Starting at 1, the client begins incrementing its local sequence number each time a Sequenced Data Packet is received. If the TCP/IP connection is ever broken, the client can then re-log into the server indicating the current session and its next expected sequence number. By doing this, the client is guaranteed to always receive every sequenced message in order, despite TCP/IP connection failures.

SoupTCPBinary also permits the client to send messages to the server using Unsequenced Data Packets at any time after the Login Accepted Packet is received. These messages may be lost during TCP/IP socket connection failures.

1.3. Heartbeats

SoupTCPBinary uses logical heartbeat packets to quickly detect link failures. The server must send a Server Heartbeat packet anytime more than 1 second has passed since the server last sent any data. This ensures that the client will receive data on a regular basis. If the client does not receive anything (neither data nor heartbeats) for an extended period of time, it can assume that the link is down and attempt to reconnect using a new TCP/IP socket.

Similarly, once logged in, the client must send a Client Heartbeat packet anytime more than 1 passes since the client last sent anything. If the server doesn't receive anything from the client for an extended period of time (typically 15 seconds), it can close the existing socket and listen for a new connection.

1.4. Data Types

Character data fields are standard ASCII bytes.

Numeric fields use ASCII digits and are padded on the left with spaces.

2. SoupTCPBinary Packet Types

1.5. Logical Packets Sent by Server

1.5.1. Login Accepted Packet

The SoupTCPBinary server sends a Login Accepted Packet in response to receiving a valid Login Request from the client. This packet will always be the first packet sent by the server after a successful login request.

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type	2	1	'A'	Login Accepted Packet
Session	3	10	Alphanumeric	Session ID of the session that is now logged into. Left padded with spaces.
Sequence Number	13	20	ASCII Numeric	The sequence number of the next Sequenced Message to be sent. Left padded with spaces.

1.5.2. Login Rejected Packet

The SoupTCPBinary server sends this packet in response to an invalid Login Request Packet from the client. The server closes the socket connection after sending the Login Reject Packet. This will be the only packet sent by the server in the case of an unsuccessful login attempt.

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type	2	1	'J'	Login Rejected Packet
Reject Reason Code	3	1	Alpha	'A' Not Authorized-invalid user/password 'S' Session invalid or not available.

1.5.3. Sequenced Data Packet

The Sequenced Data Packets act as an envelope to carry the actual sequenced data messages that are transferred from the server to the client. Each Sequenced Data Packet carries one message from the higher-lever protocol. The sequence number of each message is implied; the initial sequence number of the first Sequenced Data Packet for a given TCP/IP connection is specified in the Login Accepted Packet and the sequence number increments by 1 for each Sequenced Data Packet transmitted.

Since SoupTCPBinary logical packets are carried via TCP/IP sockets, the only way logical packets can be lost is in the event of a TCP/IP socket connection failure. In this case, the client can reconnect to the server and request the next expect sequence number and pick up where it left off.

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type	2	1	'S'	Sequenced Data Packed.
Message	3	Variable	Any	Defined by a higher-level protocol, zero length indicates no more messages available.

1.5.4. Server Heartbeat Packet

The server should send a Server Heartbeat Packet anytime more than 1 second passes where no data has been sent to the client. The client can then assume that the link is lost if it does not receive anything for an extended period of time.

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type	2	1	'H'	Server Heartbeat Packet.

1.6. Logical Packets Sent by the Client

1.6.1. Login Request Packet

The SoupTCPBinary client must send a Login Request Packet immediately upon establishing a new TCP/IP socket connection to the server.

Client and server must have mutually agreed upon the username and password fields. They provide simple authentication to prevent a client from inadvertently connecting to the wrong server.

Both Username and Password are case-insensitive and should be padded on the right with spaces.

The server can terminate an incoming TCP/IP socket if it does not receive a Login Request Packet within a reasonable period of time (typically 30 seconds).

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type	2	1	'L'	Login Request Packet.
Username	3	6	Alphanumeric	Username
Password	9	10	Alphanumeric	Password
Requested Session	19	10	Alphanumeric	Specifies session to log onto. All blanks to log onto the currently active session.
Requested Sequence Number	29	20	ASCII Numeric	Specifies the next sequence number the client wants to receive upon connection or 0 to start receiving the most recently generated message.

1.6.2. Unsequenced Data Packet

The Unsequenced Data Packets act as an envelope to carry the actual data messages that are transferred from the client to the server. These messages are not sequenced and may be lost in the event of a socket failure. The higher-level protocol must be able to handle these lost messages in the case of a TCP/IP socket connection failure.

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type	2	1	'U'	Unsequenced Data Packet.
Message	3	Variable	Any	Data defined by a higher protocol.

1.6.3. Client Heartbeat Packet

The client should send a Client Heartbeat Packet anytime more than 1 second passes where no data has been sent to the server. The server can then assume that the link is lost if it does not receive anything for an extended period of time.

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type	2	1	'R'	Client Heartbeat Packet.

1.6.4. Logout Request Packet

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

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type	2	1	'O'	Logout Request Packet.