SoupBinTCP

Version 4.10

1. Overview

SoupBinTCP 4.00 is a lightweight point-to-point protocol, built on top of TCP/IP sockets that allow delivery of a set of sequenced and unsequenced messages from a server to a client in real-time. SoupBinTCP guarantees that the client receives each sequenced message generated by the server in the correct order, even across underlying TCP/IP socket connection failures. SoupBinTCP 4.00 delivers unsequenced messages on a best-effort basis only.

SoupBinTCP clients can send messages to the server. These messages are not sequenced and may be lost in the case of a TCP/IP socket failure.

SoupBinTCP 4.00 is ideal for systems where a server needs to guarantee delivery of certain messages to a client in real-time but does not require the same level of guarantees for any other messages. The server may generate messages that are delivered only on a best-effort basis (unsequenced messages), and the client generated messages require no guarantees either because the data stream is unidirectional or because the server application generates higher-level acknowledgments for any important client-generated messages.

SoupBinTCP is designed to be used in conjunction with higher lever protocols that specify the contents of the sequenced and unsequenced messages that SoupBinTCP delivers. The SoupBinTCP protocol layer is opaque to the higher-level messages, and permits any value within its messages.

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

1.1 SoupBinTCP Logical Packets

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

Each SoupBinTCP logical packet has:

A. a two byte big-endian length that indicates the length of rest of the packet (meaning the length of the payload plus the length of the packet type – which is 1)

B. a single byte header which indicates the packet type

C. a variable length payload

Two Byte Packet	Packet	Variable-length

Length	Type	payload

SoupBinTCP Logical Packet Structure

Notes:

The SoupBinTCP 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 SoupBinTCP protocol does not define a maximum payload length.

Unlike the ASCII version, the payload may contain the line feed character or any character.

1.2 Protocol Flow

A SoupBinTCP 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 and Unsequenced 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. Note that clients must not count Unsequenced Data Packets from the server when calculating the sequence number.

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

Typically, when initially logging into a server the client will 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.

SoupBinTCP 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

SoupBinTCP uses logical heartbeat packets to quickly detect link failures. The server must send a Server Heartbeat packet anytime more than 1 second has past 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 second has passed since the client last sent anything. If the server doesn't receive anything from the client for the period of time specified in "Login Request Packet", it will close the existing socket and listen for a new connection

1.4 End of Session Marker

The server indicates that the current session has terminated by sending an End of Session Message. This indicates that there will be no more messages contained in this session.

The client will have to reconnect and login with the new Session ID or a blank(space-filled) Session ID to begin receiving messages for the next available session.

1.5 Data Types

Character data fields are standard ASCII bytes.

Integer fields are binary big-endian values.

2. SoupBinTCP Packet Types

2.1 Debug Packet

A debug packet can be sent by either side of a SoupBinTCP connection at anytime. Debug packets are intended to provide human readable text that may aid in debugging problems. Debug Packets should be ignored by both client and server application software.

Debug Packet

Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet

Packet Type	2	1	"+"	Debug Packet
Text	[3]Variable		Alpha-numeric	Free form human readable text.

2.2 Logical Packets Sent by a SoupBinTCP Server

2.2.1 Login Accepted Packet

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

Login Accepted Packet

Name	Offset	Len	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	Alpha-numeric	The session ID of the session that is now logged into. Left padded with spaces.
Sequence Number	13	20	Numeric	The sequence number in ASCII of the next Sequenced Message to be sent. Left padded with spaces.

2.2.2 Login Rejected Packet

The SoupBinTCP 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. The Login Rejected Packet will be the only non-debug packet sent by the server in the case of an unsuccessful login attempt.

Login Reject Packet

Name	Offset	Len	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	See Login Reject Codes below.

Login Reject Codes

Code	Explanation
"A"	Not Authorized. There was an invalid username and password combination in the Login Request Message.
"S"	Session not available. The Requested Session in the Login Request Packet was either invalid or not available.

2.2.3 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 sequenced 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 SoupBinTCP 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.

Sequenced Data Packet

Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet

Packet Type		2	1	"S"	Sequenced Data Packet
Message		3	Variable	Any	Defined by a higher-level protocol

2.2.4 Unsequenced Data Packet

The Unsequenced Data Packets act as an envelope to carry the unsequenced data messages that are transferred from the server to the client. Each Unsequenced Data Packet carries one unsequenced message from the higher-lever protocol. These messages are unsequenced, and must not update the client's initial sequence number for any subsequent Login attempt.

Unsequenced Data Packets from the server are delivered on a best-effort basis only. They will be delivered at most once in the same order that they are generated by the higher level protocol. However, some messages may be lost due to TCP/IP connection trouble including full buffers. If message loss occurs, no portion of the lost message will be transmitted. SoupBinTCP does not define the higher level protocol behavior when unsequenced messages are lost. These semantics are defined by the higher level protocol.

Unsequenced Data Packet

Name	Offset	Len	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	Defined by a higher-level protocol

2.2.5 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.

Server Heartbeat Packet

Name	Offset	Len	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

2.2.6 End of Session Packet

The server will send an End of Session Packet to denote that the current session is finished. The connection will be closed shortly after this packet, and the user will no longer be able to reconnect to the current session.

End of Session Packet

Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet
Packet Type	2	1	"Z"	End of Session Packet

2.3 Logical Packets Sent by the SoupBinTCP Client

2.3.1 Login Request Packet

The SoupBinTCP 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).

Login Request Packet

Name	Offset	Len	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 the session the client would like to log into, or all blanks to log into the currently active session.
Requested Sequence Number	29	20	Numeric	Specifies the next sequence number in ASCII the client wants to receive upon connection, or 0 to start receiving the most recently generated message.
Heartbeat Timeout	49	5	Numeric	Client Heartbeat Timeout in milliseconds. Failure to receive a message or a heartbeat for the specified duration causes server socket to close and listen for new connection.

2.3.2 Unsequenced Data Packets

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.

Unsequenced Data Packet

Name	Offset	Len	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	Alpha-numeric	Defined by a higher-level protocol

2.3.3 Client Heartbeat Packets

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.

Client Heartbeat Packet

Name	Offset	Len	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

2.3.4 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.

Logout Request Packet

Name	Offset	Len	Value	Notes
Packet Length	0	2	Binary	Number of bytes after this field until the next packet
Packet Type	2	1	"O"	Logout Request Packet

3. Support

Any questions about the SoupBinTCP specification should be emailed to devsupport@nasdaqomx.com.

4. Current Restrictions

None known.

5. Revision History

5.1 Version 1.00 – 9/1/1999

5.1.1 Initial distribution.

5.2 Version 2.00 - 10/29/2001

5.2.1 Added Heart beats in both directions to remove the dependence on TCP/IP keep-alive to detect link failures. Server and client are now both guaranteed to send something (either data or heartbeat) at least once per second. This way, if you don't hear anything from the socket for several seconds, you can assume the socket is dead and close it.

5.2.2 Added Debug Messages because they are handy for debugging problems. An example is to have a server send a Debug Message upon accepting a connection identifying the name of the machine. This way, someone can TELNET into a server and immediately verify that they have the right host.

5.2.3 Added "envelopes" around both the outbound Sequenced Messages and the inbound Unsequenced Message to differentiate them from the heartbeats.

5.3 Version 3.00 - 10/15/2008

5.3.1 Widened Sequence Number fields on login related packets in order to support streams of 1 billion messages or larger.

5.3.2 Added an explicit End of Session Packet so that the SoupBinTCP session layer code can more cleanly detect the end of a SoupBinTCP session.

5.3.3 Initial distribution of SoupBinTCP3.0, taken from SoupTCP3.0. Packets are two byte length-prefixed instead of new-line terminated, but otherwise the packet types are the same.

5.4 Version 3.00 – 07/20/2009

5.4.1 Removed conflicting sentence from Sequenced data packet description

5.5 Version 4.00 – 07/12/2010

5.5.1 Added unsequenced data packets from the server to the client

5.6 Version 4.1 – 01/13/2012

5.6.1 Added heartbeat timeout field to the login request packet.