Japannext JNX

SoupBinTCP Specification (All Markets)

Version 1.00

Last updated: September 5, 2025

Japannext JNX

1 Introduction

This document explains access to the **trading services** of **Japannext PTS** via the **SoupBinTCP** protocol. It provides an overview of the protocol and describes the packet types.

For further information and inquiries regarding trading services, and for questions concerning connectivity, contact Japannext Technical Support at ito@japannext.co.jp.

2 Overview

SoupBinTCP is a lightweight, point-to-point, binary messaging protocol for **guaranteed real-time delivery** of **server-to-client sequenced messages**. SoupBinTCP is widely used by financial institutions in low-latency **order entry gateways** and for low-latency **market data** feeds.

2.1 Network Stack

SoupBinTCP serves as the point-to-point transport layer for higher-level protocols such as **ITCH** and **OUCH**. In turn, SoupBinTCP uses **TCP** as its transport protocol (**Figure 1**).

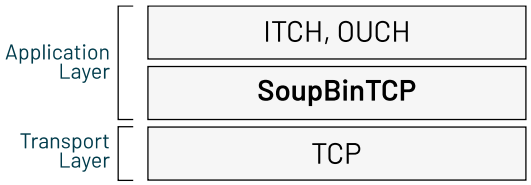

Figure 1 - SoupBinTCP stack

2.2 Core Features

Core design features of SoupBinTCP are as follows:

- Guaranteed delivery of sequenced messages in the order generated by the server.
- Recovery following a TCP/IP socket connection failure.
- Session management through login, logout, and heartbeat packets.
- Binary format with single-byte text fields not limited to printable ASCII characters.

Note: SoupBinTCP also supports client-to-server messaging but does not guarantee message delivery in the event of TCP/IP socket connection failure.

3 Logical Packets

SoupBinTCP communication is through the exchange of **logical packets**, each encapsulating a **single protocol message**. Note, however, that SoupBinTCP packets do not map directly to transport layer packets.

Each SoupBinTCP logical packet is structured as follows:

Field	Description
Length	Two bytes, big endian; denotes total packet length excluding this field.
Packet Type	Single-byte header; denotes packet type.
Data Payload	Variable length, depends on packet type; no maximum value.

4 Protocol Flow

4.1 Session Established

1. Client opens TCP/IP socket to server.
2. Client sends **Login Request Packet**.
3. If login validation is successful, server responds with **Login Accepted Packet**.
4. Server starts sending **Sequenced Data Packets**. The connection persists until the TCP/IP socket is broken or terminated.

4.2 Sequence Maintained

Sequence numbers are used to keep track of **Sequenced Data Packets**. Note, however, that the only packet to explicitly specify a sequence number is the **Login Accepted Packet**. The sequence number for subsequent packets is calculated by the client and server counting each sent and received message and incrementing a local copy of the sequence number.

Note: The first sequenced message in each session always has a sequence number of **1**, which the client increments with each new **Sequenced Data Packet** received.

In the event of a connection failure, the client is able to resume reception of sequenced messages by reconnecting to the server and specifying the following in the **Login Request Packet**:

- Session (**Requested Session** field)
- Next desired sequence number (**Requested Sequence Number** field), which is determined by referencing the maintained sequence number.

4.3 Heartbeats Sent

SoupBinTCP uses heartbeat packets to detect link failures. Both the server and the client send a heartbeat packet to each other when more than **1 second** has elapsed since the last data transmission.

If the client receives no packets of any type for a given period, the TCP/IP connection can be considered down, and accordingly, the client can attempt to reconnect to the server. Likewise, if the server has received no client packets for a given time, it can close the existing socket and continue to listen for a new connection.

Note: The current timeout value for assuming a connection failure is **15 seconds**.

4.4 Session Terminated

When the server has no further messages to send, it terminates the current session by sending a final **End of Session Message**.

5 SoupBinTCP Packet Types

5.1 Debug Packet

Debug packets provide human-readable text for troubleshooting purposes. Either side of a SoupBinTCP connection can send a debug packet at any time but should avoid processing the packet.

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until next packet.
Packet Type	2	1	'+'	Debug Packet
Text	3	Variable	Alphanumeric	Human-readable text.

5.2 Logical Packets Sent by SoupBinTCP Server

5.2.1 Login Accepted Packet

The SoupBinTCP server sends a **Login Accepted Packet** in response to a valid **Login Request Packet** received from the client. Always the first non-debug 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 next packet.
Packet Type	2	1	'A'	Login Accepted Packet
Session	3	10	Alphanumeric	Session ID of current session; left-padded with spaces.
Sequence Number	13	20	Numeric	Sequence number (ASCII) of next Sequenced Message to be sent; left-padded with spaces.

5.2.2 Login Rejected Packet

The SoupBinTCP server sends this packet in response to an invalid **Login Request Packet** from the client and then closes the socket connection. Note that the **Login Rejected Packet** is the only non-debug 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 next packet.
Packet Type	2	1	'J'	Login Rejected Packet
Reject Reason Code	3	1	Alpha	See **Table 1** below.

Table 1 - Login reject codes

Code	Explanation
A	**Not authorized**. The Login Request Packet contained an invalid Username and Password combination, or the specified TCP port did not correspond with Username.[1]
S	**Session not available**. The Login Request Packet contained an invalid or unavailable Requested Session.

[1] Japannext assigns each user a unique username and a corresponding port number—which must be used together in a SoupBinTCP session. This is relevant to clients who have been assigned multiple port numbers (due to having multiple SoupBinTCP users). In such a scenario, it's not possible to mix-and-match Usernames and port numbers.

5.2.3 Sequenced Data Packet

Sequenced Data Packets encapsulate server-to-client sequenced messages, with one message per packet.

Note: SoupBinTCP packets depend on open TCP/IP sockets for successful delivery. In the event of a TCP/IP socket connection failure, the SoupBinTCP client can resume reception of sequenced messages by reconnecting to the server and specifying the session and the next desired sequence number (or '**0**' for the most recently generated message).

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until next packet.
Packet Type	2	1	'S'	Sequenced Data Packed
Message	3	Variable	Any	Defined by a higher-level protocol.

5.2.4 Server Heartbeat Packet

The server should send a **Server Heartbeat Packet** whenever more than **1 second** has elapsed since its last data transmission. Accordingly, if the client does not receive any packets for a given time, it can assume a lost TCP/IP connection and attempt to reconnect to the server.

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

5.2.5 End of Session Packet

When the server has no further messages to send, it terminates the current session by sending a final **End of Session Message**. Thereafter, the connection is closed.

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

5.3 Logical Packets Sent by SoupBinTCP Client

5.3.1 Login Request Packet

The SoupBinTCP client must send a **Login Request Packet** immediately after establishing a new TCP/IP socket connection to the SoupBinTCP server. If the server does not receive a **Login Request Packet** within a reasonable time (typically **30 seconds**), it can terminate the incoming TCP/IP socket connection.

Note: Username and password credentials enable basic authentication that prevents the client from erroneously connecting to an unintended server.

Name	Offset	Length	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until next packet.
Packet Type	2	1	'L'	Login Request Packet
Username	3	6	Alphanumeric	Username; case-sensitive, right-padded with spaces.
Password	9	10	Alphanumeric	Password; case-sensitive, right-padded with spaces.
Requested Session	19	10	Alphanumeric	Session to log in to, or blank to log in to the currently active session; left-padded with spaces.
Requested Sequence Number	29	20	Numeric	Sequence number (ASCII) of next Sequenced Message to be sent, or 0 to for the most recently generated message; left-padded with spaces.

5.3.2 Unsequenced Data Packets

Unsequenced Data Packets encapsulate client-to-server messages, with one message per packet. Since these messages are not sequenced, a TCP/IP socket connection failure will render them unrecoverable. Therefore, the higher-level protocol should have sufficient provisions to deal with such a scenario.

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

5.3.3 Client Heartbeat Packets

The client should send a **Client Heartbeat Packet** whenever more than **1 second** has elapsed since its last data transmission. Accordingly, if the server does not receive any client packets for a given time, it can assume a lost TCP/IP connection, close the existing socket, and continue to listen for a new connection.

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

5.3.4 Logout Request Packet

The client sends a **Logout Request Packet** to the server to request connection termination. Upon receipt, the server terminates the connection and closes the associated TCP/IP socket.

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

6 Revision History

Date	Version	Description
2025-09-08	1.00	Initial version.