



J-GATE

Alternative Data Service
J-GATE ITCH Binary Data
Specification

Version 1.0

September 2021

JPX Market Innovation and Research, Inc.

We shall not be liable to any persons or entities for any disadvantages or problems whatsoever arising from the use of information contained in this document.

The contents of this document are subject to change without notice.

Revision history to version 1.0

#	Date	Chapter	Change description	Remark
1	2021/9/21	-	Initial version.	-

TABLE OF CONTENTS

1 ABOUT THIS SPECIFICATION 1

1.1 INTRODUCTION 1

2 SERVICE DETAILS 2

2.1 DATA OVERVIEW 2

3 ITCH BINARY FILE 3

3.1 FILE SPECIFICATION 3

3.2 BINARY LAYOUT 3

3.3 DATA SAMPLE 5

1 About This Specification

1.1 Introduction

This is a specification document for “J-GATE ITCH Binary Data”, one of datasets of the Alternative Data Service (“the Service”). It mainly describes information necessary for users of the Service to handle the dataset.

It is prohibited to redistribute this document.

2 Service Details

2.1 Data Overview

J-GATE ITCH Binary Data is a set of binary files containing all of market data messages disseminated via J-GATE ITCH protocol. By decoding the file according to a binary format described later, it is possible to build orderbook information of each series.

The file contains all of multicast packets disseminated from ITCH Feed except for heartbeat packets. Out of ITCH Data Feeds 1 and 2 (both active), the file provides data of Feed 1.

Please note that if there is a gap in ITCH packet sequence due to a system issue or any other reason, the message of the dropped sequence number will be missing in the file.

3 ITCH Binary File

3.1 File Specification

File name	Format
ITCH_MCG1_yyyymmdd.bin	Binary
ITCH_MCG2_yyyymmdd.bin	Binary
ITCH_MCG3_yyyymmdd.bin	Binary
ITCH_MCG4_yyyymmdd.bin	Binary
ITCH_MCG5_yyyymmdd.bin	Binary
ITCH_MCG6_yyyymmdd.bin	Binary
ITCH_MCG7_yyyymmdd.bin	Binary
ITCH_MCG8_yyyymmdd.bin	Binary

One file records data from system startup at 6:00 a.m. of the day to night session close of the following day.
Data of each multicast group is provided in an individual file.

3.2 Binary Layout

For each ITCH packet printed in the file, a MoldUDP64 protocol header is prepended with data of a UDP payload length (4 bytes).

Data of a UDP payload length is in little-endian, whereas the ensuing MoldUDP64 payload part is in network byte order (big-endian).

For details of MoldUDP64 and ITCH protocol specifications and message structures, please refer to ITCH Connectivity Manual published by Osaka Exchange, Inc.

Below is an outline of data structure corresponding to one ITCH packet. It consists of a UDP payload length and a MoldUDP64 header followed by multiple ITCH message blocks.

```
{
  "UDP payload length": uint32,
  "MoldUDP64 header": {
    "Session": char[10],
    "Sequence number": int64,
    "Message block count": int16
  },
  "ITCH message block 1": {
    "Message length": int16,
    "Message data": Each ITCH message (variable)
  },
  "ITCH message block n": {
    "Message length": int16,
    "Message data": Each ITCH message (variable)
  }
}
```


3.3 Data Sample

Below is a sample of the binary file.

	UDP data length (4 bytes)	MoldUDP64 session (10 bytes)	MoldUDP64 sequence number (8 bytes)	
00000000	14 00 00 00	4a 50 32 36 37 38 32 34 30 38	00 00JP26782408..
00000010	00 00 00 00	00 01 00 00	14 00 00 00 4a 50 32 36JP26
00000020	37 38 32 34 30 38	00 00 00 00 00 00 00 00	01 00 00	782408.....
00000030	14 00 00 00	4a 50	MoldUDP64 message block count (2 bytes) 38 00 00JP26782408..
00000040	00 00 00 00	00 01 00 00	14 00 00 00 4a 50 32 36JP26
00000050	37 38 32 34 30 38	00 00 00 00 00 00 00 00	01 00 00	782408.....
00000060	14 00 00 00	4a 50 32 36 37 38 32 34 30 38	00 00JP26782408..
00000070	00 00 00 00	00 01 00 00	14 00 00 00 4a 50 32 36JP26
00000080	37 38 32 34 30 38	00 00 00 00 00 00 00 00	01 00 00	782408.....
00000090	14 00 00 00	4a 50 32 36 37 38 32 34 30 38	00 00JP26782408..
000000a0	00 00 00 00	00 01 00 00	14 00 00 00 4a 50 32 36JP26
000000b0	37 38 32 34 30 38	00 00 00 00 00 00 00 00	01 00 00	782408.....