



INVESTORS EXCHANGE DEEP SNAP SPECIFICATION

Version 1.0

Updated: January 13th, 2021



Table of Contents

| | |
|--|----------|
| OVERVIEW | 3 |
| TRANSPORT PROTOCOL OPTIONS | 3 |
| ARCHITECTURE | 4 |
| DATA TYPES | 4 |
| NETWORK DETAILS | 5 |
| DEEP SNAP TCP/IP | 5 |
| Snapshot Configuration | 5 |
| DEEP SNAP Server Addresses | 5 |
| MESSAGE FORMATS | 6 |
| SnapshotRequest Message (MessageType 'r') | 6 |
| ErrorResponse Message (MessageType 'e') | 7 |
| Reject Reason Code | 7 |
| Snapshot Response | 8 |
| SnapshotStart Message (MessageType 's') | 8 |
| SnapshotData Message (MessageType 'd') | 8 |
| SnapshotEnd Message (MessageType 'x') | 8 |
| DEEP Feed messages in Snapshot Response | 9 |



OVERVIEW

Participants of Investors Exchange (“IEX” or the “Exchange”) may use DEEP to receive real-time depth of book quotations and last sale information direct from IEX. DEEP also supports several security-related administrative messages and provides event controls, such as start of day and end of day, to participants.

The depth of book quotations received via DEEP provide an aggregated size of resting displayed orders at a price and side, and do not indicate the number or size of individual orders at any price level. Non-displayed orders and non-displayed portions of reserve orders are not represented in DEEP. DEEP also provides last trade price and size information. Trades resulting from either displayed or non-displayed orders matching on IEX are reported. Routed executions are not reported.

DEEP provides short sale restriction status, trading status, operational halt status, and security event information via security-related administrative messages. Lastly, DEEP provides event information about the market and data feed via administrative messages.

Please refer to [IEX DEEP Specification](#) for more specific DEEP feed information.

The proposed DEEP SNAP service is intended to augment the current DEEP gap-fill retransmission service by adding a separate snapshot protocol to allow consumers to accelerate late-start recovery. It does not modify the existing DEEP feed or retransmission protocols.

TRANSPORT PROTOCOL OPTIONS

DEEP SNAP will use a new request-response TCP/IP unicast protocol.

Note: Snapshot Responses contain data messages detailed in the [IEX DEEP Specification](#) and the [IEX Transport specification](#).



ARCHITECTURE

DEEP SNAP is a request-response protocol with no unsolicited messages (i.e. heartbeats). Applications would start to receive and buffer DEEP multicast before connecting to a DEEP SNAP server on a well-known port and would issue a SnapshotRequest Message.

In response the DEEP SNAP server will either provide a Snapshot Response or ErrorResponse Message.

A Snapshot Response provides a point-in-time snapshot of the DEEP Price level data and trading status data for all IEX Symbols along with the associated DEEP feed Sequence number.

On receipt of a Snapshot Response the applications would initialize their internal Price level books for each Symbol using data from the Snapshot Response, then apply any buffered real-time updates with higher sequence numbers.

Applications would then disconnect from the DEEP SNAP server and continue to process the DEEP feed real-time multicast as normal.

Connections may be timed-out and disconnected by either client or server if no data has been transmitted for a significant period (10 seconds).

DATA TYPES

These are identical to the DEEP Specification:

- String: Fixed-length ASCII byte sequence, left justified and space filled on the right
- Long: 8 bytes, signed integer
- Price: 8 bytes, signed integer containing a fixed-point number with 4 digits to the right of an implied decimal point
- Integer: 4 bytes, unsigned integer
- Byte: 1 byte, unsigned integer
- Timestamp: 8 bytes, signed integer containing a counter of nanoseconds since POSIX (Epoch) time UTC
- Event Time: 4 bytes, unsigned integer containing a counter of seconds since POSIX (Epoch) time UTC

All binary fields are in **little endian** format.

Note that each byte is represented by two hexadecimal digits in the examples within this specification.



NETWORK DETAILS

DEEP SNAP TCP/IP

Snapshot Configuration

- Supported Retransmission Protocol(s): TCP
- Maximum Requests: Quota restricted (1000/day)
- Supported Request Type(s): SnapshotRequest

DEEP SNAP Server Addresses

| SITE | XC Type | Server | Port | Credentials |
|---------------------------------------|---------------|----------------|-------|--------------------|
| IEX POP (Equinix NY5) | Primary (A) | 23.226.155.163 | 11379 | Contact Market Ops |
| | Secondary (B) | 23.226.155.227 | 11379 | Contact Market Ops |
| Disaster Recovery (Equinix CH4) | Tertiary (C) | TBD | TBD | TBD |
| IEX Testing Facility (Equinix NY5) | ITF (I) | 23.226.155.19 | 33379 | “TEST:password” |

Please contact marketops@iextrading.com to procure DEEP SNAP credentials.

Please contact itfsupport@iextrading.com for support help related to the ITF.



MESSAGE FORMATS

SnapshotRequest Message (MessageType 'r')

The SnapshotRequest message is sent from the Client to the DEEP SNAP Server to authenticate and request a Snapshot Response.

| Field Name | Offset | Length | Type | Description/Notes |
|-------------------------|--------|--------|---------|--|
| Message Length | 0 | 2 | Integer | Length of message in bytes not including this field (41) |
| Message Type | 2 | 1 | Byte | 'r' (0x72) - SnapshotRequest |
| Authentication Token | 3 | 40 | Byte | Token supplied by IEX MarketOps. Left justified string <i>user:password</i> space padded on right. |
| ChannelID | 43 | 4 | Integer | Channel Identifier. |
| SessionID | 47 | 4 | Integer | Session Identifier. |
| Minimum Sequence Number | 51 | 8 | Long | Minimum Sequence Number useable by client |

Total Message Data length is 59 bytes.

Applications will normally wait until they receive and start buffering DEEP Feed Multicast messages before connecting and sending a SnapshotRequest to the DEEP SNAP Server. The SnapshotRequest's ChannelID, SessionID and Minimum Sequence Number fields should be populated with values from the DEEP FEED IEX TP-Header in the Multicast packet.



ErrorResponse Message (MessageType 'e')

Sent by the DEEP SNAP server to the client when a SnapshotRequest is rejected.

| Field Name | Offset | Length | Type | Description/Notes |
|--------------------|--------|--------|---------|---|
| Message Length | 0 | 2 | Integer | Length of message in bytes not including this field (2) |
| Message Type | 2 | 1 | Byte | 'e' (0x65) - ErrorResponse |
| Reject Reason Code | 3 | 1 | Byte | Reason the SnapshotRequest was rejected |

Total Message Data length is 4 bytes.

Reject Reason Code

'A' (0x41) - Authentication Failure

'C' (0x43) - Incorrect ChannelID in SnapshotRequest

'E' (0x45) - Snapshot Request already active

'Q' (0x51) - Quota Exceeded

'R' (0x52) - Snapshot Not Yet Available (no snapshot \geq requested Minimum Sequence Number is available)

'S' (0x53) - Incorrect SessionID in SnapshotRequest

'U' (0x55) - Snapshot Service Temporarily Unavailable



Snapshot Response

A Snapshot Response is sent from the DEEP SNAP server to the client when a client [SnapshotRequest](#) is successful. A Snapshot Response consists of a [SnapshotStart](#) message, followed by [SnapshotData](#) messages, and concluded by a [SnapshotEnd](#) message.

SnapshotStart Message (MessageType 's')

| Field Name | Offset | Length | Type | Description/Notes |
|----------------|--------|--------|---------|--|
| Message Length | 0 | 2 | Integer | Length of message in bytes not including this field (9) |
| Message Type | 2 | 1 | Byte | 's' (0x73) - SnapshotStart |
| SnapshotLength | 3 | 8 | Long | Length in bytes of the complete Snapshot Response . Includes this SnapshotStart message, all SnapshotData messages and the SnapshotEnd message |

SnapshotData Message (MessageType 'd')

| Field Name | Offset | Length | Type | Description/Notes |
|----------------------------|--------|--------|----------|--|
| Message Length | 0 | 2 | Integer | Length of message in bytes not including this field (variable) |
| Message Type | 2 | 1 | Byte | 'd' (0x64) - SnapshotData |
| IEX-TP Header | 3 | 40 | Byte | see IEX Transport specification |
| IEX-TP MessageBlock Length | 43 | 2 | Integer | see IEX Transport specification |
| IEX-TP Message Data | 45 | | Variable | see IEX Transport specification and IEX DEEP Specification |

The IEX-TP Message Data will contain messages in [IEX DEEP](#) format and may include any DEEP message types.

SnapshotEnd Message (MessageType 'x')

| Field Name | Offset | Length | Type | Description/Notes |
|--------------------------|--------|--------|---------|---|
| Message Length | 0 | 2 | Integer | Length of message in bytes not including this field (9) |
| Message Type | 2 | 1 | Byte | 'x' (0x78) - SnapshotEnd |
| Snapshot Sequence Number | 3 | 8 | Long | Sequence at which the Snapshot was created |



DEEP Feed messages in Snapshot Response

The IEX-TP Message Data returned in the SnapshotData messages will contain DEEP Feed messages required to rebuild the current state of the DEEP Feed books for each symbol at the SequenceNumber the Snapshot was created.

Snapshot Content:

1. [SnapshotStart](#) message
2. For each IEX Listed symbol (at time of snapshot)
 - a) Latest **SecurityDirectory** Message
3. Latest **SystemEvent** message (if exists at time of snapshot)
4. For each symbol (at time of snapshot)
 - a) Latest **TradingStatus** Message (if exists at time of snapshot)
 - b) Latest **SecurityEvent** Message (if exists at time of snapshot)
 - c) Latest **OperationalHaltStatus** Message (if exists at time of snapshot)
 - d) Latest **ShortSalePriceTestStatus** Message (if exists at time of snapshot)
 - e) Latest **OfficialPrice** Message (if exists at time of snapshot)
 - f) All **PriceLevelUpdate** Messages (that are active at time of snapshot)
5. [SnapshotEnd](#) Message (includes [SnapshotSequenceNumber](#))

Note: All DEEP Feed messages (shown in **bold**) are wrapped within a DEEP SNAP [SnapshotData](#) Message that includes an IEX-TP Header providing DEEP Feed sequencing and timing information.

Note: All **PriceLevelUpdate** Messages will have **EventFlags=1** regardless of the original value sent on the DEEP Feed. All messages will add a new level to the symbol's orderbook. No messages will delete or modify levels.

Note: **PriceLevelUpdate** Messages will not be in the original DEEP Feed Sequence or SendTime order.