

Historical Price Data Messages

- [HISTORICAL_PRICE_DATA_REQUEST](#)
- [HISTORICAL_PRICE_DATA_RESPONSE_HEADER](#)
- [HISTORICAL_PRICE_DATA_REJECT](#)
- [HISTORICAL_PRICE_DATA_RECORD_RESPONSE](#)
- [HISTORICAL_PRICE_DATA_TICK_RECORD_RESPONSE](#)
- [Providing Bid Volume and Ask Volume in Historical Data](#)

HISTORICAL_PRICE_DATA_REQUEST [s_HistoricalPriceDataRequest structure] Client >> Server

This is a message from the Client to the Server for requesting historical price data.

This request can be on the same or a separate network socket connection compared to the streaming market data. This is going to be specified by the Server.

Field Name	Field Description
[unsigned int16] Size	The standard message size field by constructor.
[unsigned int16] Type	The standard message type field by constructor.
[int32] RequestID	Unique integer identifier to identify historical price data response m Server will contain this identifier matched up with the request from identifier only needs to be unique price data messages. It can continue used with other classes of messages

[char] Symbol	The Symbol historical price data is
[char] Exchange	Optional: The exchange for the Sy
[HistoricalDataIntervalEnum] RecordInterval	<p>The interval/timeframe of each r range of the historical data request</p> <ul style="list-style-type: none"> • INTERVAL_TICK = 0 • INTERVAL_1_SECOND • INTERVAL_1_MINUTE • INTERVAL_1_DAY = 86 • INTERVAL_1_WEEK =
[t_DateTime] StartDateTime	<p>The starting Date-Time for the hi returned, if available for the specifi</p> <p>If it is not set or set to 0, then this Server to return data starting at available for the Symbol.</p>
[t_DateTime] EndDateTime	<p>The ending Date-Time for the hi returned.</p> <p>If it is not set or set to 0, then this Server to return data ending at th available for the symbol.</p>

<p>[unsigned int32] MaxDaysToReturn</p>	<p>MaxDaysToReturn specifies the number of days of data the Server needs to be available, counting back from the last data available for the symbol, or EndTime if it is set to a value.</p> <p>If MaxDaysToReturn is set to 0, then the Server.</p>
<p>[unsigned int8] UseZLibCompression</p>	<p>Set this to 1, to request the compression in the response when available. The Server can optionally ignore support compression or does not support compression for any reason.</p> <p>The HISTORICAL PRICE DATA REQUEST will not be compressed. Only the first 100 records themselves.</p> <p>When receiving a batch of data from the socket, give it to ZLib. It will return uncompressed data and you put them through process as many complete messages out of it that you can. Then process until finished.</p>
<p>[unsigned int8] RequestDividendAdjustedStockData</p>	<p>In the case of a stock symbol, set to 1 will request dividend adjusted data from the Server, if available. It is optional support this.</p>
<p>[unsigned int16] Integer_1</p>	<p>A general purpose 2 byte flag field for use by the Server which can be used for anything the Client and Server require.</p>

HISTORICAL_PRICE_DATA_RESPONSE_HEADER

[s_HistoricalPriceDataResponseHeader structure] Server >> Client

When a historical price data request is not rejected, this message header will begin the historical price data response from the Server. There will be one

HISTORICAL_PRICE_DATA_RESPONSE_HEADER message sent ahead of the

[HISTORICAL_PRICE_DATA_RECORD_RESPONSE](#) /

[HISTORICAL_PRICE_DATA_TICK_RECORD_RESPONSE](#) messages. If the

NoRecordsToReturn field is nonzero, then there are no further records that will be sent by the Server in response to the request by the Client.

This message is never compressed.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The numeric identifier from the historical price data request that this response is in response to.
[HistoricalDataIntervalEnum] RecordInterval	The data interval of type HistoricalDataIntervalEnum requested by the Client.
[unsigned int8] UseZLibCompression	1 = All subsequent messages are using standard ZLib compression. 0 = no compression.

[unsigned int8] NoRecordsToReturn	If there are no records to return in response to the request and there was no error, this will be set to 1.
[float] IntToFloatPriceDivisor	This field is no longer used.

HISTORICAL_PRICE_DATA_REJECT [s_HistoricalPriceDataReject structure] Server >> Client

When the Server rejects a historical price data request from the Client, a **HISTORICAL_PRICE_DATA_REJECT** message will be sent.

This message is never compressed.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.

[int32] RequestID	<p>The numeric identifier from the historical price data request that this response is in response to.</p>
[char] RejectText	<p>Text reason for rejection.</p>
[HistoricalPriceDataRejectReasonCodeEnum] RejectReasonCode	<p>Integer identifier identifying the reason for the rejection. For the text reason, refer to the RejectText field.</p>

<p>[unsigned int16] RetryTimeInSeconds</p>	<p>This is an optional field from the Server. This field will normally be zero.</p> <p>If a retry is intended to be performed, the server may give an indication of how long to wait in seconds. This field indicates that.</p> <p>This field is not recommended to be used. If it is used, it is really an indication of a substandard Server.</p>
---	---

HISTORICAL_PRICE_DATA_RECORD_RESPONSE **[s_HistoricalPriceDataRecordResponse structure] Server >> Client**

The **HISTORICAL_PRICE_DATA_TICK_RECORD_RESPONSE** message is used when the **RecordInterval** field in a historical data request message is set to a value greater than **INTERVAL_TICK**. For example, if the **RecordInterval** is **INTERVAL_1_MINUTE**, then a message of this type will contain data for a 1 minute timeframe with a start time specified by the **StartDateTime** field.

Even when **RecordInterval** is **INTERVAL_TICK**, the **HISTORICAL_PRICE_DATA_TICK_RECORD_RESPONSE** message can still be used instead of [HISTORICAL_PRICE_DATA_TICK_RECORD_RESPONSE](#).

This message can be part of a compressed series of messages of this same type, if the Client requested compression be used.

Field Name	Field Description

[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The numeric identifier from the historical price data request that this response is in response to.
[t_DateTime] StartDateTime	<p>The starting Date-Time in UTC of the data record in this message.</p> <p>It is part of the DTC Protocol specification that this must be the <u>starting</u> Date-Time of the data record.</p>
[double] OpenPrice	The Open price of the data record in this message.
[double] HighPrice	<p>The High price of the data record in this message.</p> <p>In the case where NumTrades is 1, the HighPrice field can be the Ask/Offer price at the time of the trade. In this case the OpenPrice field needs to be 0 in this case.</p>
[double] LowPrice	<p>The Low price of the data record in this message.</p> <p>In the case where NumTrades is 1, the LowPrice field can be the Bid price at the time of the trade. In this case the OpenPrice field needs to be 0 in this case.</p>

[double] LastPrice	<p>The Last price of the data record in this message.</p>
[double] Volume	<p>The Volume of this data record of this message.</p>
(union) [unsigned int32] OpenInterest [unsigned int32] NumTrades	<p>The Open Interest or Number of Trades of this data record in this message.</p>
[double] BidVolume	<p>The volume of trades at the bid price or lower of the data record in this message.</p> <p>In the case where this message consists of a single trade, if the trade was at the Ask, then BidVolume must be zero.</p> <p>This is the volume of the trades, where the aggressor of the trade was the seller.</p>
[double] AskVolume	<p>The volume of trades at the ask price or higher of the data record in this message.</p> <p>In the case where this message consists of a single trade, if the trade was at the Bid, then AskVolume must be zero.</p> <p>This is the volume of the trades, where the aggressor of the trade was the buyer.</p>
[unsigned int8] IsFinalRecord	<p>Set to 1 to indicate final record in response to the historical price data request.</p> <p>The default is 0 meaning there are more records to follow.</p>

HISTORICAL_PRICE_DATA_TICK_RECORD_RESPONSE **[s_HistoricalPriceDataTickRecordResponse structure] Server >> Client**

This is the response message when the **RecordInterval** field in a historical data request message is set to **INTERVAL_TICK**.

If the Server does not support 1 Tick historical data or does not have 1 Tick historical data for the specified time period, it can respond with [HISTORICAL_PRICE_DATA_RECORD_RESPONSE](#) messages instead. The Server must only respond with messages of one type in response to a particular historical price data request.

This message can be part of a compressed series of messages of this same type, if the Client requested compression be used.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The numeric identifier from the historical price data request that this response is in response to.
[t_DateTimeWithMilliseconds] DateTime	The Date-Time of the trade.
[AtBidOrAskEnum] AtBidOrAsk	This indicates whether the trade occurred at the Bid price or lower or at the Ask price or higher.

[double] Price	The price of the trade.
[double] Volume	The volume of the trade.
[unsigned int8] IsFinalRecord	<p>Set to 1 to indicate final record in response to the historical price data request.</p> <p>The default is 0 meaning there are more records to follow.</p>

Providing Bid Volume and Ask Volume in Historical Data

There are different ways of providing volume of trades that have occurred at the Bid price, and volume of the trades that have occurred at the Ask price, in the historical price data response messages.

One way of doing this, is to use the [HISTORICAL_PRICE_DATA_RECORD_RESPONSE](#) message to provide tick by tick data. Follow the instructions below.

- Set HISTORICAL_PRICE_DATA_RECORD_RESPONSE::OpenPrice to 0.
- Set HISTORICAL_PRICE_DATA_RECORD_RESPONSE::HighPrice to the Ask/Offer price at the time of the trade.
- Set HISTORICAL_PRICE_DATA_RECORD_RESPONSE::LowPrice to the Bid price at the time of the trade.
- Set HISTORICAL_PRICE_DATA_RECORD_RESPONSE::LastPrice to the trade price.
- Set HISTORICAL_PRICE_DATA_RECORD_RESPONSE::NumTrades to 1.
- Set HISTORICAL_PRICE_DATA_RECORD_RESPONSE::Volume to the volume or quantity of the trade.
- Set HISTORICAL_PRICE_DATA_RECORD_RESPONSE::BidVolume to the volume of the trade, if it occurred at the Bid. If you want to let the Client make this determination of whether the trade occurred at the bid or ask, then set this to 0.
- Set HISTORICAL_PRICE_DATA_RECORD_RESPONSE::AskVolume to the volume of the trade, if it occurred at the Ask. If you want to let the Client make this determination of whether the trade occurred at the bid or ask, then set this to 0.

Example of how fields need to be set in the HISTORICAL_PRICE_DATA_RECORD_RESPONSE message.

- DTC::t_DateTimeWithMicrosecondsInt StartDateTime = 1640995200000000 (2022-1-1

- 00:00:00UTC)
- double OpenPrice = 0
 - double HighPrice = 101.8
 - double LowPrice = 101.7
 - double LastPrice = 101.8
 - double Volume = 10
 - uint32_t NumTrades = 1
 - double BidVolume = 0
 - double AskVolume = 0
 - uint8_t IsFinalRecord = 0

The above will mean the DTC client will set the AskVolume to 10.

*Last modified Thursday, 26th May, 2022.