

# Data and Trading Communications (DTC) Protocol Server

- [Introduction](#)
- [Server Usage](#)
- [Special Notes](#)
- [Historical Price Data Server](#)
- [US Bond Futures Pricing](#)
- [Using DTC Server for Data and Trading in Another Sierra Chart Instance](#) (New Page)
- [Troubleshooting Connection Problems to DTC Server](#)
- [Restrictions](#)
- [Web Socket Connections](#)

## Introduction

---

Sierra Chart provides a full [DTC Protocol](#), server for market data and for trading. This is a TCP/IP socket based server.

Multiple client connections are supported. The connection can be from any IP address and this is controllable within Sierra Chart.

With the DTC Protocol Server, all trading related messages are fully supported. Both non-simulated and simulated trading is supported when using the DTC Protocol Server. For proper support for both of these, make sure you are running a current version of Sierra Chart.

All of the encoding methods supported by the DTC Protocol are supported by the DTC Protocol Server in Sierra Chart.

The DTC Protocol historical data server port now supports all DTC Encodings as of version 1558.

## Server Usage

---

Follow the instructions below to use the DTC Protocol server. Both a standard simple socket connection and also websocket connections are supported.

1. Select **Global Settings >> Sierra Chart Server Settings** in Sierra Chart.
2. Enable **DTC Protocol Server >> Enable DTC Protocol Server**.
3. In the **Listening Port** box, specify the port number for the server to listen on or keep the default. This cannot be 0.
4. Enable **DTC Protocol Server >> Allow Trading** for the trading related DTC Protocol messages to be supported. Otherwise, they will not be by the DTC Server in Sierra Chart.

5. To have Sierra Chart automatically send a [SECURITY\\_DEFINITION\\_RESPONSE](#) message when Sierra Chart begins to track a symbol, then enable the **Auto Send Security Definition for New Symbols** option. In this particular case, the **RequestID** field will be set to **110000000** (This is not binary, but base 10).
6. Set the **Encoding** setting to the encoding that you intend to use instead of **Automatic** unless you will be using the [Encoding Request Sequence](#).
7. Set the **Allowed Incoming IPs** setting to one of the following:
  - **Local Computer Only**: Only connections originating from the local computer are allowed.
  - **Local Subnet Only**: Only connections originating on the local sub network are allowed.
  - **Any IP**: A connection originating from any IP address is allowed.
8. Enable the **Require Authentication** option if you want to require that a Username and Password be specified in the Logon message from the Clients. This is required to be enabled and will be automatically enabled when **Allowed Incoming IPs** is set to **Any IP**. The Username and Password must match the Sierra Chart Username and Password used to log into the server instance of Sierra Chart.
9. Press **OK** to save the settings and close the settings window.
10. Establish a TCP/IP network connection to **127.0.0.1** using the port number that was specified in the **Listening Port** box. This is going to be a simple standard network connection.

Effective with version 1798 or higher of Sierra Chart, a websocket connection is also now supported. When a websocket connection is used, the encoding will always be JSON Compact. For example, you can use Python or JavaScript to establish a websocket connection and then send null terminated JSON DTC messages and receive JSON DTC messages. There is nothing special to do other than this. A websocket type of connection is automatically detected.

In no case should you ever need to send a [DTC Encoding Request](#) when using a websocket connection because it will not be properly interpreted unless it is binary encoded.

11. Follow the DTC Protocol [Connection and Logon Sequence](#). If you have set the **Encoding** to a setting other than **Automatic**, then there is no need to perform the Encoding Request Sequence.
12. The Username and Password in the [LOGON\\_REQUEST](#) message should be blank unless

**Global Settings >> Sierra Chart Server Settings >> DTC Protocol Server >> Require Authentication** is enabled. A Username and Password will also be required when **Global Settings >> Sierra Chart Server Settings >> DTC Protocol Server >> Allowed Incoming IPs** is set to **Any IP**.

The Username and Password will need to be set to the same ones that were used to log into the Sierra Chart instance which is used as a server.

These are the only fields which are used in the [LOGON\\_REQUEST](#) message:

**Username, Password, HeartbeatIntervalInSeconds, ClientName, TradeMode, ProtocolVersion (automatically set), Integer1.**

13. For the most up-to-date market data messages, set Integer1 in the Logon Request message with the following flags using a bitwise or operator:
  - DTC\_LOGON\_REQUEST\_INTEGER\_1\_SUPPORT\_UNBUNDLED\_TRADES  
= 0x4
  - DTC\_LOGON\_REQUEST\_INTEGER\_1\_USE\_MARKET\_DEPTH\_UPDATE\_FLOAT\_WITH\_  
= 0x80
  - DTC\_LOGON\_REQUEST\_INTEGER\_1\_SUPPORT\_MARKET\_DEPTH\_SNAPSHOT\_LEVEL  
= 0x800
  - DTC\_LOGON\_REQUEST\_INTEGER\_1\_SUPPORT\_MARKET\_DATA\_UPDATE\_TRADE\_VOLUME  
= 0x80000
  - DTC\_LOGON\_REQUEST\_INTEGER\_1\_SUPPORT\_MARKET\_DATA\_UPDATE\_BIDASK\_SPREAD  
= 0x100000
14. Sierra Chart always uses the latest DTC Protocol version.
15. For detailed information to work with the DTC Protocol and the supported messages and fields, refer to the [DTC Message and Procedures](#) page.
16. The Sierra Chart DTC Server does not use the **Exchange** field for orders , market data or historical market data in any of the DTC Protocol messages.

You only need to set the **Symbol** field to the exact symbol as it is used within the Sierra Chart main program itself. The symbols can be found in **File >> Find Symbol** within Sierra Chart.

17. The following market data messages are used by the Sierra Chart DTC Protocol Server and no others.
  - [MARKET\\_DATA\\_REQUEST](#)
  - [MARKET\\_DATA\\_REJECT](#)
  - [MARKET\\_DATA\\_SNAPSHOT](#)
  - [MARKET\\_DATA\\_UPDATE\\_TRADE\\_COMPACT](#)
  - [MARKET\\_DATA\\_UPDATE\\_TRADE\\_WITH\\_UNBUNDLED\\_INDICATOR](#)
  - [MARKET\\_DATA\\_UPDATE\\_LAST\\_TRADE\\_SNAPSHOT](#)
  - [MARKET\\_DATA\\_UPDATE\\_BID\\_ASK\\_COMPACT](#)
  - [MARKET\\_DATA\\_UPDATE\\_SESSION\\_OPEN](#)
  - [MARKET\\_DATA\\_UPDATE\\_SESSION\\_HIGH](#)
  - [MARKET\\_DATA\\_UPDATE\\_SESSION\\_LOW](#)
  - [MARKET\\_DATA\\_UPDATE\\_SESSION\\_VOLUME](#)
  - [MARKET\\_DATA\\_UPDATE\\_OPEN\\_INTEREST](#)
  - [MARKET\\_DATA\\_UPDATE\\_SESSION\\_SETTLEMENT](#)
  - [MARKET\\_DEPTH\\_REQUEST](#)
  - [MARKET\\_DEPTH\\_REJECT](#)
  - [MARKET\\_DEPTH\\_SNAPSHOT\\_LEVEL](#)
  - [MARKET\\_DEPTH\\_UPDATE\\_LEVEL\\_COMPACT](#)

---

## Special Notes

When modifying a Stop-Limit order using [CANCEL\\_REPLACE\\_ORDER](#) or

[CANCEL\\_REPLACE\\_ORDER\\_INT](#), then only Price1 can be modified. Price2 will be set to a price to maintain the same price difference that Price2 had to Price1 before the order modification.

The [SYMBOLS\\_FOR\\_EXCHANGE\\_REQUEST](#) message is supported by the DTC Server. However, the exchange field is ignored and Sierra Chart returns security definitions for the known symbols for the current Data/Trade service that Sierra Chart is currently set to work with.

The Sierra Chart DTC Server does not use the **Exchange** field for orders or market data in any of the DTC Protocol messages. You only need to set the **Symbol** field to the exact symbol as it is used within the Sierra Chart main program itself. The symbols can be found in **File >> Find Symbol** within Sierra Chart.

To receive market depth data update messages which contain milliseconds in the **DateTime** field, it is necessary to set [LogonRequest.Integer\\_1](#) to 0x80 when logging into the DTC Server. This should be combined with a [bitwise or](#) operator with any other flags set into Integer\_1. The message structure for market depth updates is **s\_MarketDepthUpdateLevelFloatWithMilliseconds**. The **DateTime** field is the standard UNIX time format representing milliseconds as an integer. This is effective as of version 1949.

## Historical Price Data Server

---

The DTC Protocol server in Sierra Chart has a full DTC compatible Historical Price Data Server.

Follow the instructions below to use this historical data server.

1. Select **Global Settings >> Sierra Chart Server Settings >> DTC Protocol Server**.
2. Enable the **Enable DTC Protocol Server** option.
3. Set the **Historical Data Port** number. The default is 11098. Historical data is downloaded through this port. The **Listening Port** does not support historical data.
4. Press **OK**.
5. Follow the [Historical Price Data](#) procedures section for using DTC messages with the server.
6. This DTC Historical Price Data server only accepts one request per connection. You will notice that **s\_LogonResponse::OneHistoricalPriceDataRequestPerConnection** is set to 1. After the first request is served, another one will not be accepted and the network socket connection will eventually get closed.
7. All of the [encodings](#) are supported. It is not recommended to use JSON encoding for large amounts of historical Intraday data.

As of version 2100, the websocket protocol is also supported.

8. Historical price data is served from existing Historical Daily and Intraday data files. If the file is not up to date or does not exist, the data will first be downloaded and then after served back. A connection to the data feed will be made in Sierra Chart if it is necessary, to download the missing data. Therefore, only the data that is present in the file can be served.

To increase the amount of Intraday data available for a particular symbol, the

#### **Global Settings >> Data/Trade Service Settings >> Maximum Historical Intraday Days to Download**

settings need to be increased. After increasing either of these settings, it is necessary to open an Intraday chart for the symbol in Sierra Chart and then re-download the data with **Edit >> Delete All Data and Download**.

9. The `Flag_1` member in the [HISTORICAL\\_PRICE\\_DATA\\_REQUEST](#) request message can be set to 0x10 to always force a remote download from the specified Start Date-Time. You can use a bitwise or to combine this with any other parameters in this flag variable.

Version 1682 or higher is necessary for support for the `HISTORICAL_PRICE_DATA_REQUEST::EndDateTime` field. This field is not supported when requesting historical data with a time period of 1 day and greater.

When `HISTORICAL_PRICE_DATA_REQUEST::MaxDaysToReturn` is set to a value greater than zero and the `HISTORICAL_PRICE_DATA_REQUEST::RecordInterval` is less than `DTC::INTERVAL_1_DAY`, then `MaxDaysToReturn` is followed and the starting Date-Time of data is going to be as follows: If `HISTORICAL_PRICE_DATA_REQUEST::EndDateTime` is nonzero, then it is calculated as `HISTORICAL_PRICE_DATA_REQUEST::EndDateTime` minus `HISTORICAL_PRICE_DATA_REQUEST::MaxDaysToReturn`. Otherwise, it is calculated as the Date-Time of the last record available minus `HISTORICAL_PRICE_DATA_REQUEST::MaxDaysToReturn`.

In no case will data be returned earlier than is available for the symbol.

The historical price data server implemented by Sierra Chart is intended to always to return data up to the very latest data available. Therefore, as a general rule you should only specify `HISTORICAL_PRICE_DATA_REQUEST::StartDateTime` or leave it at 0 to get all available data.

Data that has not yet been written to the Intraday data file will not be returned. So therefore, the [Intraday File Flush Time in Milliseconds](#) does have an effect upon what data is returned.

The starting Date-Time of each record for records less than 1 Day, will be the Date-Time of the first trade within the timeframe of the record.

The `HISTORICAL_PRICE_DATA_REQUEST::RecordInterval` field is the number of seconds for the data record. The Sierra Chart Historical Data Server does support using anywhere from 0, for tick by tick data, up to 86400. You are not limited to the predefined constants for this field.

In the case of requesting historical Intraday data, since the data is served from Intraday data files and these files can contain records which are a variable amount of time anywhere from 1 Tick/Trade to 1 Minute, all records are served with the message type

**HISTORICAL\_PRICE\_DATA\_RECORD\_RESPONSE** in all cases. In the case of a single tick record, the **OpenPrice** will be 0. The **HighPrice** will be the Ask price at the time of the trade. The **LowPrice** will be the Bid price at the time of the trade. The **NumTrades** field will be 1.

## **Requesting and Processing Summary Trade Flags**

---

For CME futures data, it is supported to request the flags which indicate the beginning and ending trades of a summary trade. This can be used to reconstruct the summary trade.

The beginning and ending trade of a summary trade is specified through the Open field of the Historical price data record. The following are the values:

- const float SINGLE\_TRADE\_WITH\_BID\_ASK = 0.0F;
- const float FIRST\_SUB\_TRADE\_OF\_UNBUNDLED\_TRADE = - 1.99900095e+37F;
- const float LAST\_SUB\_TRADE\_OF\_UNBUNDLED\_TRADE = - 1.99900197e+37F;

To request these particular values be set, when making a historical data request you need to set:

- HistoricalPriceDataRequest.Integer\_1 |= 0x8;

---

## US Bond Futures Pricing

---

The [Market Data Messages](#) that Sierra Chart sends for market data, use 4-byte floats for numbers.

This is sufficient accuracy for all symbols except for US bond market symbols.

In this particular case, the precision is sufficient enough to convert to the actual value. This can be done by dividing the given float value by the **MinPriceIncrement** from the [SECURITY\\_DEFINITION\\_RESPONSE](#) message for the symbol, rounding that result to the nearest integer and multiplying that integer by **MinPriceIncrement**.

---

## Using DTC Server for Data and Trading in Another Sierra Chart Instance

---

Refer to the [Using DTC Server for Data and Trading in Another Sierra Chart Instance](#) documentation.

---

## Troubleshooting Connection Problems to DTC Server

---

When using an instance of Sierra Chart acting as a server by having the **DTC Protocol Server** enabled, you are connecting to it from a different computer or a different operating system instance on the same physical machine, and you encounter a problem connecting, first review the instructions in the [Using DTC Server for Data and Trading in Another Sierra Chart Instance](#) section.

Make sure the DTC server in the server instance of Sierra Chart is enabled. Check that you have correctly specified the correct IP address and Port number in the client instance of Sierra Chart which you are connecting from.

If there still is a connection problem connecting from the client instance of Sierra Chart to the server instance of Sierra Chart, then most likely this is going to be the result of the connection getting blocked somewhere between the two instances of Sierra Chart.

The specific reason for the connection failure, is going to be shown in the **Window >> Message Log** of the client instance. The operating system network socket connection error is given. You will need to conduct your own research on the meaning of that error.

The typical reason for the connection failure are firewalls and Network Address Translation (NAT) devices.

The first step is to disable the operating system (Windows) firewall to see if the connection then succeeds. If that resolves the problem, then add a rule to the Windows firewall to allow incoming connections to the Sierra Chart program. Otherwise, the resolution of a problem like this is not within the scope of Sierra Chart support.

In the case when the following error is encountered in the **Window >> Message Log** after about 20 seconds when connecting to the DTC server:

**Windows error code 10060: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.**

Then this is an indication that a firewall or NAT is either blocking or not routing the connection. You may have to use port forwarding through your NAT device.

If the

**Global Settings >> >> Sierra Chart Server Settings >> DTC Protocol Server >> Allowed Incoming IPs** setting is blocking the connection, then you will know this is happening because you will see a message like the following in the **Window >> Message Log** of the server instance of Sierra Chart:

**DTC Protocol Server | Rejecting connection from (IP address)**

## Restrictions

---

Due to legal requirements imposed currently or in the future by exchanges, data providers, or any other market data relationship Sierra Chart has, the DTC Protocol Server may be changed to not provide market data from certain exchanges, or the user of the data may have to explain and state their use to ensure compliance and if it is not compliant, the functionality will be disabled for that user or users.

This can go into effect at any point in time based upon current legal requirements which we are obliged to comply with.

It is not possible to request or use streaming market from an IP other than on the local machine. This restriction can be removed for custom purposes [upon request](#) for approved and documented purposes which are approved by the exchanges to be used.

It is not possible to access real-time or historical data from the CME Group of exchanges, EUREX or NASDAQ, US equities data originating from UTP or CTA from the DTC Protocol server.

## Web Socket Connections

---

The [DTC protocol server](#) in Sierra Chart support websocket connections. A websocket connection will automatically be used if the connected client is using the websocket protocol. There is no special configuration required.

When using a websocket connection the encoding is automatically set to [JSON Compact](#). Although only market data messages use the compact JSON format. Otherwise, the standard [JSON](#) format is used.

When connecting to the DTC Protocol Server in Sierra Chart using a websocket connection from a web browser, it is necessary to install a valid TLS certificate for the domain used. Both the private key and the certificate need to be placed in the [/ServerCertificate](#) folder within the Sierra Chart folder. The file names need to be: **TLSCertificate.key** (Private key), **TLSCertificate.crt** (Certificate).

Open SSL is being used, and the above files are specified to these Open SSL functions respectively: `SSL_CTX_use_PrivateKey_file`, `SSL_CTX_use_certificate_file`.

Each message that your websocket client sends, needs to end with a null terminator character (1 byte with value of 0).

---

\*Last modified Friday, 24th March, 2023.