

Data and Trading Communications (DTC) Protocol Test Client

- [Introduction](#)
- [Test Client Instructions](#)
- [Notes](#)
- [Populating Symbol List in Sierra Chart](#)

Introduction

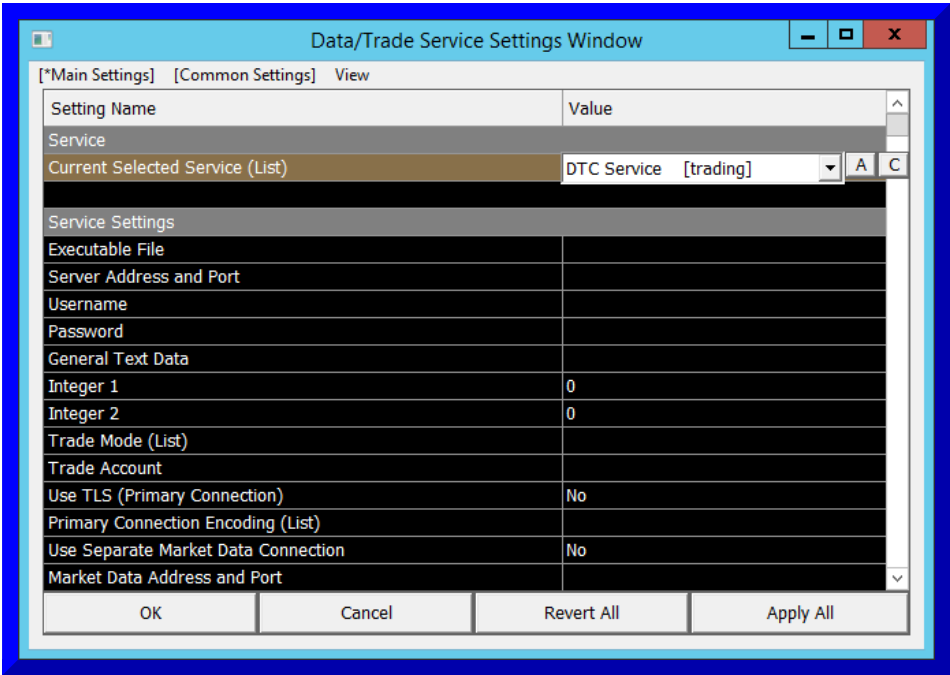
For a test Client for [DTC Protocol](#) Server development, use the Sierra Chart charting and trading platform.

Refer to the [Test Client Instructions](#) below.

Sierra Chart also has a full [DTC Protocol Server](#) within the Sierra Chart program. Therefore, Sierra Chart can be both used as a Test Client and Server for the DTC Protocol.

Test Client instructions

Sierra Chart is a fully compatible DTC Client. For Server developers that would like to use a test Client, Sierra Chart is available at no charge.



To use Sierra Chart as a DTC Protocol test Client, follow these instructions:

1. [Create a Sierra Chart account](#) if you do not already have one.
2. [Download and install](#) Sierra Chart.
3. If you are developing a DTC Server for general public use, then contact **[Java Script is required to view the email address. Enable Java Script]** with your Sierra Chart Account Name and request that your account be activated for free access for DTC Protocol development.
4. [Log into Sierra Chart](#) with your Sierra Chart Account Name and Password.
5. Select **Global Settings >> Data/Trade Service Settings** on the Sierra Chart menu.
6. Set the **Service** to **DTC Service**. This fully supports the DTC Protocol as a Client and can be used to establish a connection to a DTC Protocol compliant Server for testing.
7. Set **Connection Details >> Server Address** to the IP and port number of the DTC Server. Format: **Address:Port**. Multiple addresses can be separated by a comma (,) and at every connection the next address in the list will be used.

The Port number is required when connecting to a remote server. The port number is optional when connecting to a local server which is started as an executable program. If a port is specified, that will be a port which is used to connect to the Server whether it is local or remote. If the Port is not specified, then Sierra Chart will determine a Port when connecting to a local server executable program and pass it to the program on the commandline as a parameter. Example: **/Port:1234**.

8. The Sierra Chart Test Client supports non-TLS and TLS connections. TLS connections use [Open SSL](#). Set **Connection Details >> Use TLS** to True to use a TLS connection. TLS version 1.2 is used. This is supported in version 1295 and higher of Sierra Chart.
9. The Sierra Chart Test Client does not support ZLib compression for historical price data on the primary DTC Protocol connection used for market data or trading. It is necessary to use a separate historical data connection for compression. In this case set **Use Separate Historical Data Connection** to **True** to use a separate connection.
10. Set the **Primary Connection Encoding** to the encoding that you want to use for the primary connection.

If the DTC Server is going to only support one encoding and both the client and server can agree ahead of time on that encoding, then enable **Skip Encoding Request for Primary Connection** (version 1769 and higher). In this case, the server does not need to support the **DTC Encoding Request** message.

Otherwise, it is necessary that the Server has implemented support for the [Encoding Request](#) message.

11. If the server does not require a logon message, then enable **Skip Logon Request for Primary Connection**. This is supported in version 1769 and higher.
12. Set the other **Service Settings** as required.
13. **Connection Details >> Executable File**: If you want Sierra Chart to run a server executable file upon connection, then specify the path and filename to the server executable in the **Connection Details >> Executable File** box. The Server program will need to listen for a network connection on the specified Port.

The format is: **[absolute or relative path to executable server program]/[filename.exe]**.

14. Press **OK**.
15. To connect to the DTC Server, select **File >> Connect to Data Feed**.
16. If the DTC Server has set **TradingIsSupported** to a nonzero number in the LOGIN_RESPONSE message, then after 5 seconds, the DTC Server will receive the following request messages from the Client: [OPEN_ORDERS_REQUEST](#), [HISTORICAL_ORDER_FILLS_REQUEST](#), [CURRENT_POSITIONS_REQUEST](#).

[HISTORICAL_ORDER_FILLS_REQUEST](#) is only sent once upon the connection and will not be sent again during the connection to the Server. The other messages can be received by the Server at any later time during the connection.

You can cause them to be sent by selecting **Trade >> Refresh Trade Data from Service**.

17. To open a chart **File >> Find Symbol**. Manually type a symbol into the **Selected Symbol** box and press **Open Intraday Chart** or **Open Historical Chart**. This will cause the following messages to be sent to the DTC server:
 1. [SECURITY_DEFINITION_FOR_SYMBOL_REQUEST](#)
 2. [MARKET_DATA_REQUEST](#)
 3. [HISTORICAL_PRICE_DATA_REQUEST](#)
18. To perform trading, disable Trade Simulation by removing the checkmark from **Trade >> Trade Simulation Mode On**. For further instructions, refer to [Overview of Trading](#).
19. Refer to the [Trade Window](#) documentation to submit an order.

Notes

At this time the **s_HistoricalPriceDataResponseTrailer** message is not yet supported in Sierra Chart by either the Server or the Client.

It is recommended in **Global Settings >> Data/Trade Service Settings >> Service Settings** for the DTC Server, that you specify a separate connection for historical data.

This will download data on a separate thread. So you need to handle historical data with a separate network socket on your server side. This is really the only proper way to process large amounts of historical data. Otherwise, the interface of Sierra Chart can freeze when processing historical data.

In Sierra Chart when you have set **Global Settings >> Data/Trade Service Settings >> Primary Connection Always Supports Historical Data** to True, then whether the message field **s_LogonResponse::HistoricalPriceDataSupported** is set to true1 or not, historical data will be considered to be still supported by the primary server connected to.

Populating Symbol List in Sierra Chart

It is supported to add symbols that a DTC Server provides market data and trading support for, to Sierra Chart so they can be found by the user. And so that the settings for those symbols are known by Sierra Chart when Sierra Chart is started, before Sierra Chart connects to the DTC Server.

This is supported by sending an unsolicited [SECURITY_DEFINITION_RESPONSE](#) message to Sierra Chart for each symbol you want to add to Sierra Chart.

This can be done after the Logon process from the Client to the Server is complete.

For each security definition that is sent unsolicited to Sierra Chart, this will cause the symbol and the associated settings to be added to **Global Settings >> Symbol Settings**.

The symbols will be listed in the **Find Symbol** window which is opened through **File >> Find Symbol**.

When sending the Security Definition, set the fields of that message as follows:

- **RequestID**: This must be set to 0 to cause the Security Definition data to be inserted into the Global Symbol Settings.
- **Symbol**: This needs to be set to the symbol.
- **Exchange**: Sierra Chart combines the Symbol and Exchange into one text string by using a delimiter. Format example: **Symbol-Exchange**. If the **Exchange** field is set, then set the delimiter you want Sierra Chart to use through the **SymbolExchangeDelimiter** field in the [LOGON_RESPONSE](#) message.
- **Description**: This is recommended to be set.
- **MinPriceIncrement**: This needs to be set.
- **PriceDisplayFormat**: This needs to be set.
- **CurrencyValuePerIncrement**: This is recommended to be set.
- **IsFinalMessage**: Set this to 0. For the final security definition sent, set this to 1. If there is only one security definition sent, then set this to 1.
- **OrderIntPriceMultiplier**: Not used.
- **MarketDataIntPriceDivisor**: Not used.
- **UnderlyingSymbol**: Not used.

*Last modified Friday, 21st January, 2022.