

UDP Interface

- [General Information](#)
- [Opening Chart or Activating a Chart](#)
- [Opening or Activating a Chartbook](#)
- [Release Study DLL](#)
- [Allow Load Study DLL](#)
- [Release All Study DLLs](#)
- [Allow Load All Study DLLs](#)
- [Exit Program](#)
- [Requesting a Chart Image for Image Serving](#)
- [UDP Program Code Example](#)

General Information

Sierra Chart uses a UDP socket interface to receive short text messages to perform a few simple actions which are documented below.

All of this can be done from an external program using this interface.

To enable the UDP interface and to change the port number, select

Global Settings >> Sierra Chart Server Settings. Set the **UDP Port** to a nonzero number to enable the UDP interface.

To use the UDP interface, create a UDP socket in a separate program and send a text message to port set in the prior step at the IP address where Sierra Chart is running. Usually this will just be the localhost which is **127.0.0.1**. Refer to the [UDP Program Code Example](#).

Using UDP protocol is quite simple and supported by most any programming language.

Opening Chart or Activating a Chart

If you want to open a chart for a symbol or activate an existing chart for a particular symbol, then send a text string to the UDP port using the following format:

Symbol.dly (for Historical Daily charts)

Symbol.scid (for Intraday charts)

Examples

ESZ15.scid

\$INX.dly

Opening or Activating a Chartbook

Opening or Activating a Chartbook

If you want to open a Chartbook or activate an existing open Chartbook, then use the following UDP request string:

OPEN_CHARTBOOK--*Chartbook Name.cht*

Example

OPEN_CHARTBOOK--MyChartbook.cht

Release Study DLL

To release a custom study DLL and prevent it from being loaded again, use the following UDP request string:

RELEASE_DLL--*[DLLPathAndFileName]*

Allow Load Study DLL

To allow future loading of a custom study DLL, use the following UDP request string:

ALLOW_LOAD_DLL--*[DLLPathAndFileName]*

Release All Study DLLs

To release all custom study DLLs and prevent them from being loaded again until **ALLOW_LOAD_ALL_DLLS** is sent, use the following UDP request string:

RELEASE_ALL_DLLS

Allow Load All Study DLLs

To allow future loading of all custom study DLLs, use the following UDP request string:

ALLOW_LOAD_ALL_DLLS

Exit Program

To exit from Sierra Chart, use the following UDP string:

EXIT

This causes Sierra Chart to go through the same process as the menu command **File >> Exit**.

Requesting a Chart Image for Image Serving

It is supported to open a chart or activate an existing chart and save an image of that chart to a file.

One application for this interface is to use Sierra Chart as chart image server for a website. Sierra Chart is very capable of running in a server environment and it is extensively used by us on our servers.

For this command, use the following UDP request format:

IMAGE_REQUEST--*Symbol.Extension--Width--Height--NumberOfBars--OutputPathandFileName*

Each element in this message is separated with -- characters.

Definition of each of the message elements:

- **IMAGE_REQUEST**: This is the actual text string command to use to make an image request.
- **Symbol.Extension**: This is the symbol that you want to open a chart for. The extension indicates the type. The file extension can be **dly** (Historical Daily chart) or **scid** (Intraday chart). Example **ABC.scid**
- **Width**: The width of the image in pixels.
- **Height**: The height of the image in pixels.
- **Number of Bars**: The number of chart bars in the image. The spacing of the bars in the chart will be adjusted to meet this specification.
- **OutputPathandFileName**: The output path and file name for the image file. The output file extension determines the format. The file extension can be BMP, JPG or PNG. PNG is recommended because it is lossless and compact.

Example

IMAGE_REQUEST--ES#.dly--640--480--20--C:/temp/image.png

UDP Program Code Example

The following is an example to send a UDP command to Sierra Chart. You need to replace the **SendBuf** text string with the particular command you want to use.

Example

```

#include <stdio.h>
#include "winsock2.h"

void main()
{
    WSADATA wsaData;
    SOCKET SendSocket;
    sockaddr_in RecvAddr;
    int Port = 22903;

    int BufLen = 128;
    char SendBuf[BufLen] = "IMAGE_REQUEST--ES#.dly--640--480--20--C:\\temp\\image.png";

    //-----
    // Initialize Winsock
    WSASStartup(MAKEWORD(2,2), &wsaData);

    //-----
    // Create a socket for sending data

    SendSocket = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);

    //-----
    // Set up the RecvAddr structure with the IP address of
    // the receiver (in this example case "127.0.0.1")
    // and the specified port number.
    RecvAddr.sin_family = AF_INET;
    RecvAddr.sin_port = htons(Port);

    RecvAddr.sin_addr.s_addr = inet_addr("127.0.0.1");

    //-----
    // Send a datagram to the receiver

    printf("Sending a datagram to the receiver...\n");
    sendto(SendSocket,
        SendBuf,
        BufLen,
        0,
        (SOCKADDR *) &RecvAddr,
        sizeof(RecvAddr));

    //-----
    // When the application is finished sending, close the socket.
    printf("Finished sending. Closing socket.\n");
    closesocket(SendSocket);

    //-----
    // Clean up and quit.

    printf("Exiting.\n");
    WSACleanup();
    return;
}

```

*Last modified Wednesday, 22nd February, 2023.