OneChain Trading Suite - v1.0

# Introduction

This manual explains the many features available in the OneChain Trading Suite software.

The OneChain Trading Suite (“OCTS”) is built from proprietary technology in advanced programming languages (Microsoft C# for speed and Python for data science). OCTS is designed to trade financial futures, crytpocurrencies, or a combination of both.



**NOTE:** Throughout this document, various files are displayed that are proprietary (historical data files, back-test trade data, automated trading strategies, etc.) Any display or reference to such proprietary files is for explanatory purposes only and is not meant to express or imply these files are included as part of the OCTS software.

In terms of cryptocurrency trading, OCTS version 1.0 is optimized to work with the the APIs (Application Programming Interfaces) of the following exchanges:

1. Bitstamp
2. Bitfinex
3. Kraken
4. ItBit
5. GDAX
6. Bittrex
7. BitFlyer
8. Poloniex
9. Binance

The OCTS infrastructure is also connected (again via exchange-specific APIs) to the following cryptocurrency exchanges:

1. B2C2
2. BitMEX
3. Bitsquare
4. Bithumb
5. BnZ
6. Bleutrade
7. Blinktrade
8. BTCC
9. BTER
10. Cex
11. Changelly
12. CHBTC
13. Coinigy
14. Coinone
15. EtherDelta
16. GateIO
17. HitBTC
18. Huobi
19. Korbit
20. Kucoin
21. OkCoin
22. OKEx
23. Vaultoro
24. Wex
25. Xapo
26. XCrypto

Additionally, OCTS connects with the following non-exchange services to pull crypto-related data:

1. BitcoinCharts
2. BitcoinFees
3. BitcoinWatch
4. BitGo
5. BlockchainInfo
6. BraveNewCoin
7. Coinbase
8. Coindesk
9. CoinMarketCap
10. CaoinTracking
11. CryptoCoinCharts
12. CryptoCompare
13. Cryptonator
14. Etherchain
15. TradeBlock

In particular, OCTS interfaces with TradeBlock to allow for streaming of cryptocurrency quotes to authenticated brokers and larger retail clients.

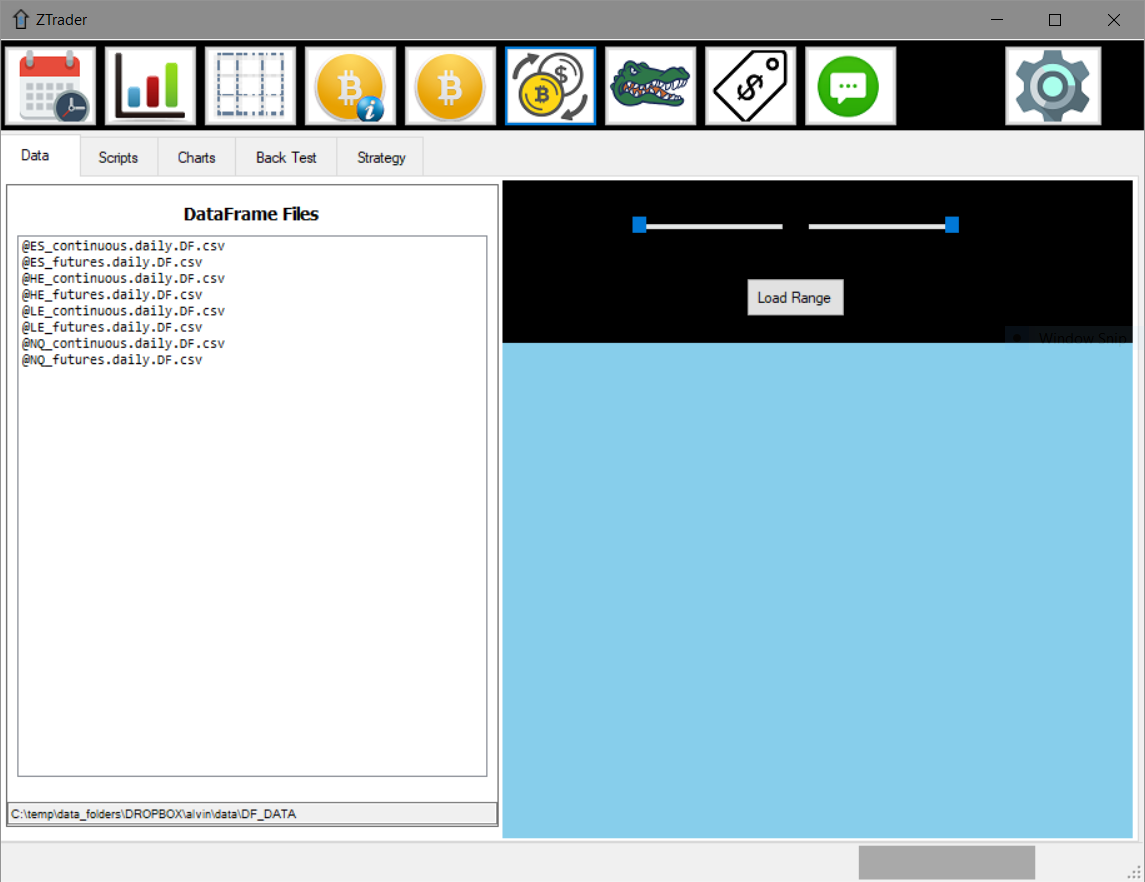
**NOTE:** This document relates to version 1.0 of OCTS. Significant UI (User Interface) improvements are underway with an ETA of 2-3 weeks. When these UI improvements are completed, this document will be updated to reflect these updates.



Figure 1 - Version 2.0 OneChain Trading Suite user interface

# Main Form

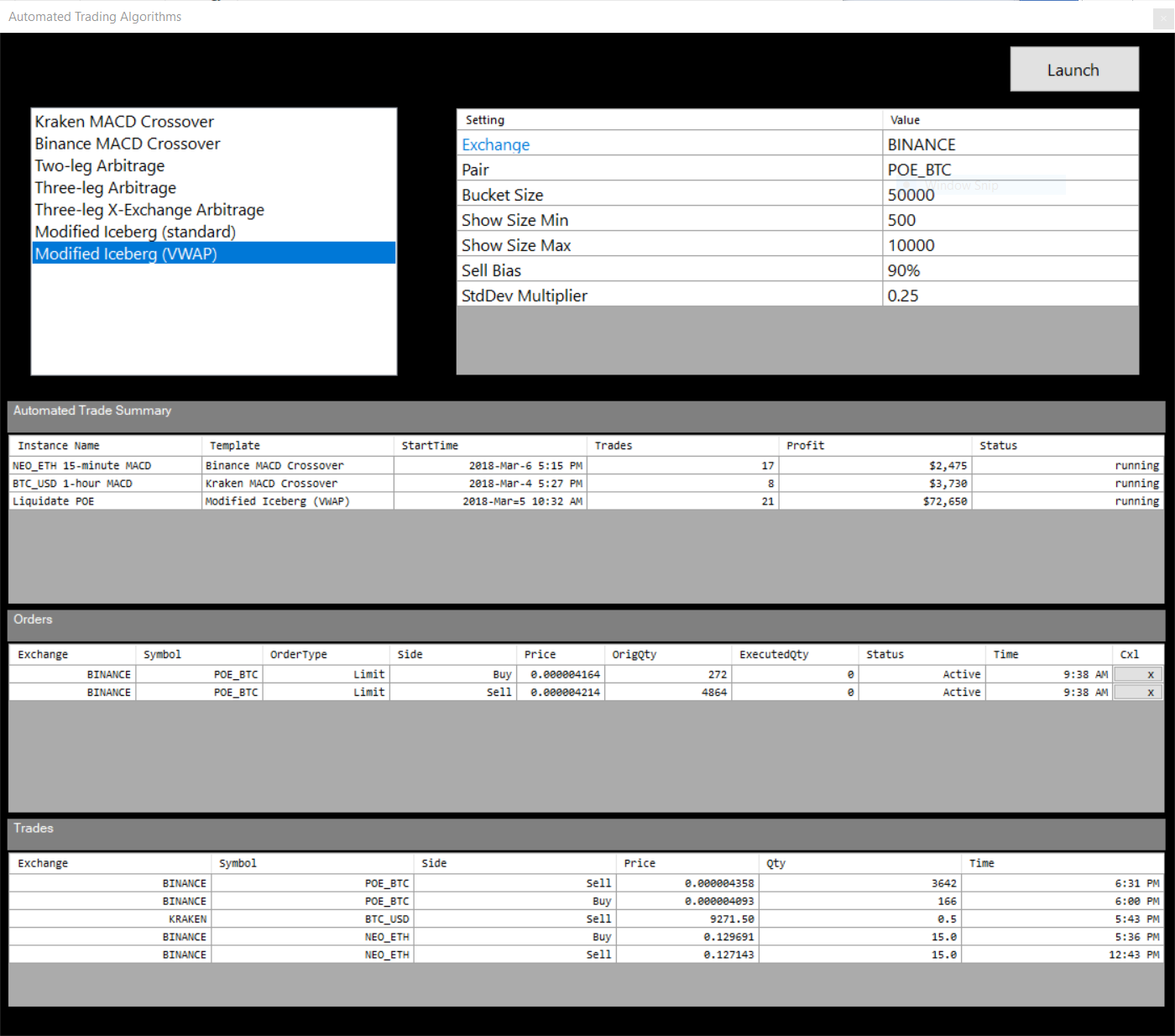
The Main OCTS form provides the ability to launch the various OCTS sub-forms, each dealing with specific technology areas:



Most OCTS functional areas (and their related forms) are launched from this Main form. For a breakdown of the sub-areas (tabs) within the Main form, see the subsequent Main Form section in this document.

# Automated Trading Algorithms Form

The Automated Trading Algorithms form allows the OCTS user to launch and monitor automated (algorithmic, or “algo”) trading strategies across multiple exchanges:



In this example, we see the settings available for the “Modified Iceberg (VWAP)” algo. The OCTS user can adjust these settings and click the “Launch” button to start the selected automated trading strategy.

The bottom of the Automated Trading Algorithms form displays sections for monitoring running algos, orders, and trades. Automated trades can be paused or stopped by right-clicking on a running algo and selecting the appropriate context menu item.

The variety and capabilities of automated trading strategies is essentially limited only by the imagination of the OCTS user. These automated trades are implemented in C# code. And although they require an experienced C# coder for their initial development, the ability to provide user-modified settings and launch/monitor these strategies ensures that any OCTS user can utilize even the most powerful automated trading algorithms.

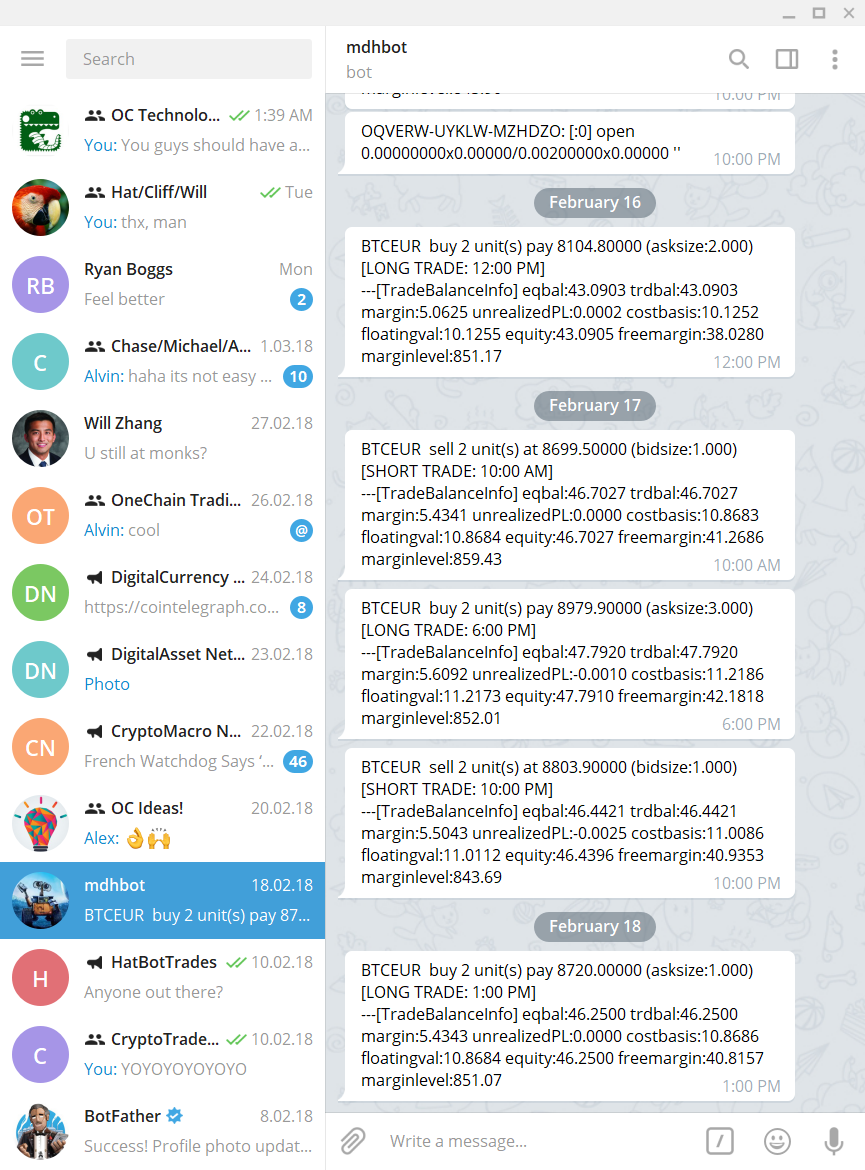
## Trade Notification

When a trade occurs in one of the currently running automated trades, the user can be informed in any or all of the following ways:

1. Telegram message
2. SMS text message
3. Via Growl/Prowl (free apps available for Android, iOS, and various desktop operating systems)
4. Standard Windows notifications (not available on Windows version prior to Windows 10)

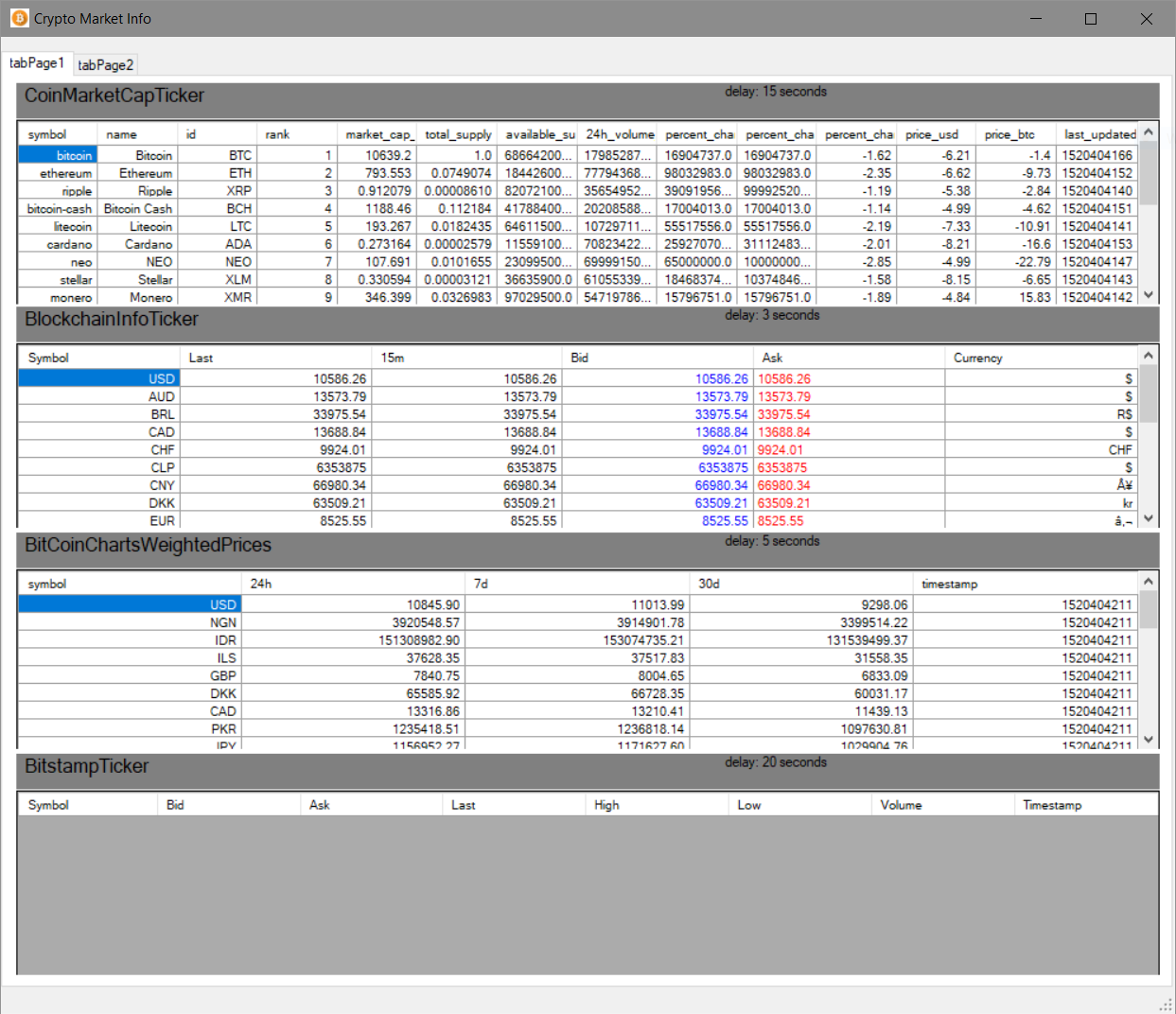
By choosing one or more of these notification methods, an OCTS user (or a group of users) will be notified whenever an automated trade occurs. By receiving notifications (even via mobile phone), an OCTS user can monitor automated trade performance without the need to constantly watch the OCTS app.

Here is a sample of trade notifications received via Telegram message:



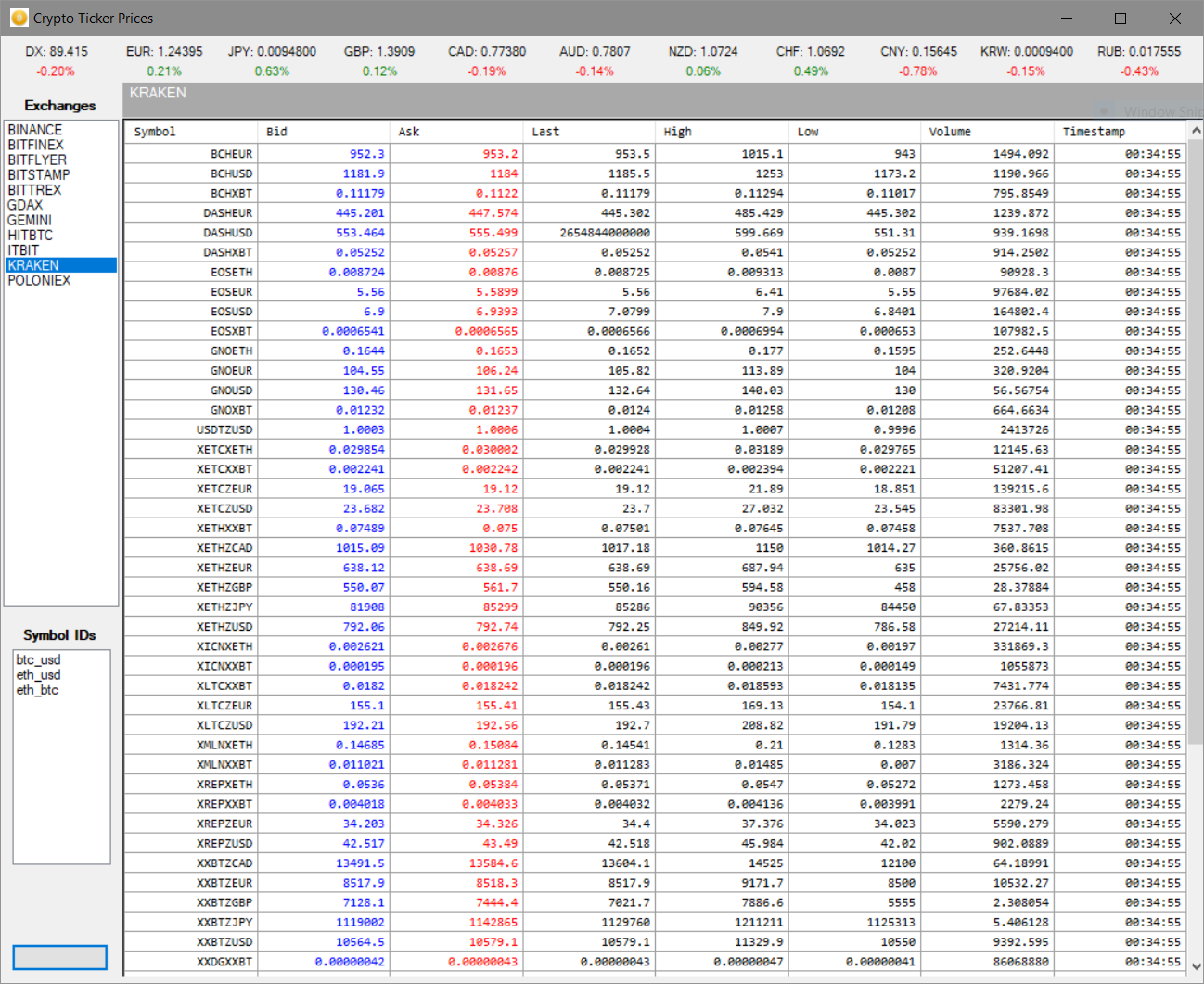
# Crypto Market Information Form

The Crypto Market Information form is the location for general (non-price-specific) information about the cryptocurrency markets. Currently, the Crypto Market Information form displays some market cap and ranking data from CoinMarketCap and some blockchain-specific information. But ultimately, this form will encompass a much larger collection of crypto market information (mining difficulty, positive/negative mentions on social media, etc.)



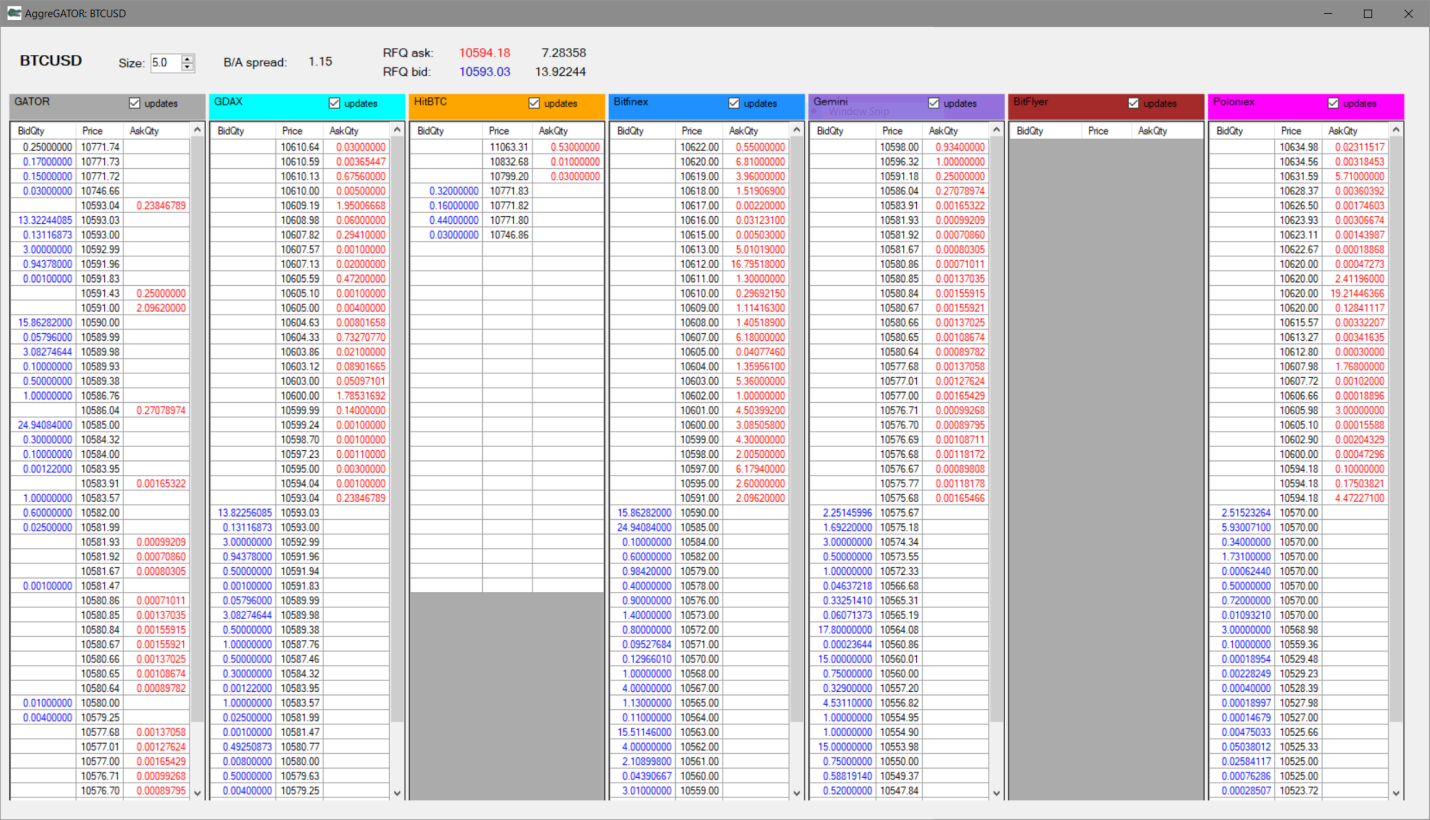
# Crypto Ticker Prices Form

Most of OCTS is designed to be used with specific cryptocurrency pairs. The Crypto Ticker Prices form, on the other hand, is designed to display a snapshot of *all* cryptocurrency pairs traded on a specific exchange. Clicking an exchange from the list on the left of the form will display a snapshot of ticker (bid/ask/last) prices for all currency pairs traded on that exchange:



# Crypto Price Aggregator Form (“GATOR”)

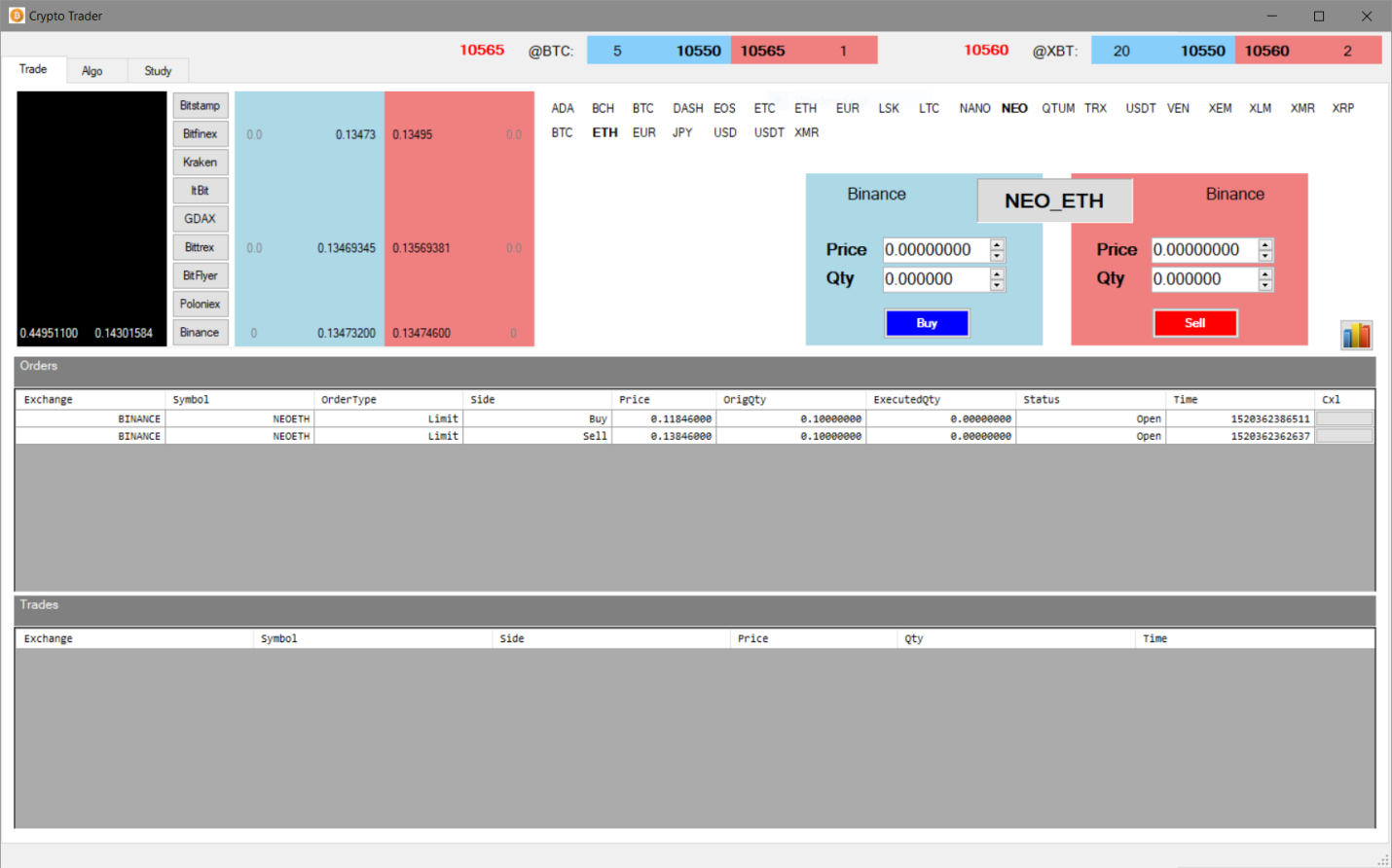
The OCTS Price Aggregator form scans multiple crypto exchanges in real-time to determine the best bid/ask prices for a given instrument. GATOR aggregates the volume available on these multiple exchanges to provide the highest current bid (buy price) and ask (sell price) for a given cryptocurrency pair. GATOR then allows the user to buy or sell against this set of exchanges/orders with a single Buy or Sell button click:



By aggregating prices across multiple exchanges, GATOR allows OCTS users to provide the best possible price to clients, customers, or internal trading strategies.

# Crypto Trader Form

To manage individual orders (across exchanges and cryptocurrency pairs), OCTS provides the Crypto Trader form:



There is a lot going on in the Crypto Trader form, so let’s break it down a bit:

* At the very top of the form, live futures prices are displayed for both the CME (@BTC) and CBOE (@XBT) futures contracts.
* Selecting a currency pair is accomplished by clicking a currency in the top row of symbols (“NEO” in this case) and a currency in the bottom row of symbols (“ETH” in this case). Once a currency pair is selected, the blue/red areas to the right of the exchange buttons display a live market (but only for those exchanges on which the selected currency pair trades). In this example, NEO\_ETH is the currency pair selected, and Bitfinex, Bittrex, and Binance show real-time prices because NEO\_ETH trades on these three exchanges.
* Additionally, the black area to the left of the exchange buttons displays the real-time balances in each currency that is part of the selected currency pair. In this case, there is a balance in both NEO (left-hand column) and ETH (right-hand column) for the Binance exchange only. If the user has balances in these cryptocurrencies on the other exchanges, those would also be displayed.
* The Buy and Sell boxes allow the user to quickly adjust price and quantity and submit either a buy or sell order to the selected exchange. (Click one of the exchange buttons to route your order to a specific exchange.)
* Finally, the bottom portion of the form displays Orders and Trades, each in their own pane. Live orders (top-most of the two lower panes) can be cancelled by clicking the right-most button in a specific grid row. Completed trades are displayed in the bottom-most of the two lower panes.

The Crypto Trader form will likely *not* be the primary interface utilized by OCTS users. Primary user interaction will likely occur through aggregated bulk order submission (via “GATOR”) or trades submitted by automated trading algorithms (see *Automated Trading Algorithms* Form below). Still, the Crypto Trader form provides an efficient interface to submit single orders and to monitor trades and monitor/modify/cancel orders submitted via these other means.

# Main Form

This section revisits the OCTS Main form and the functionality contained within its various sub-areas (tabs).

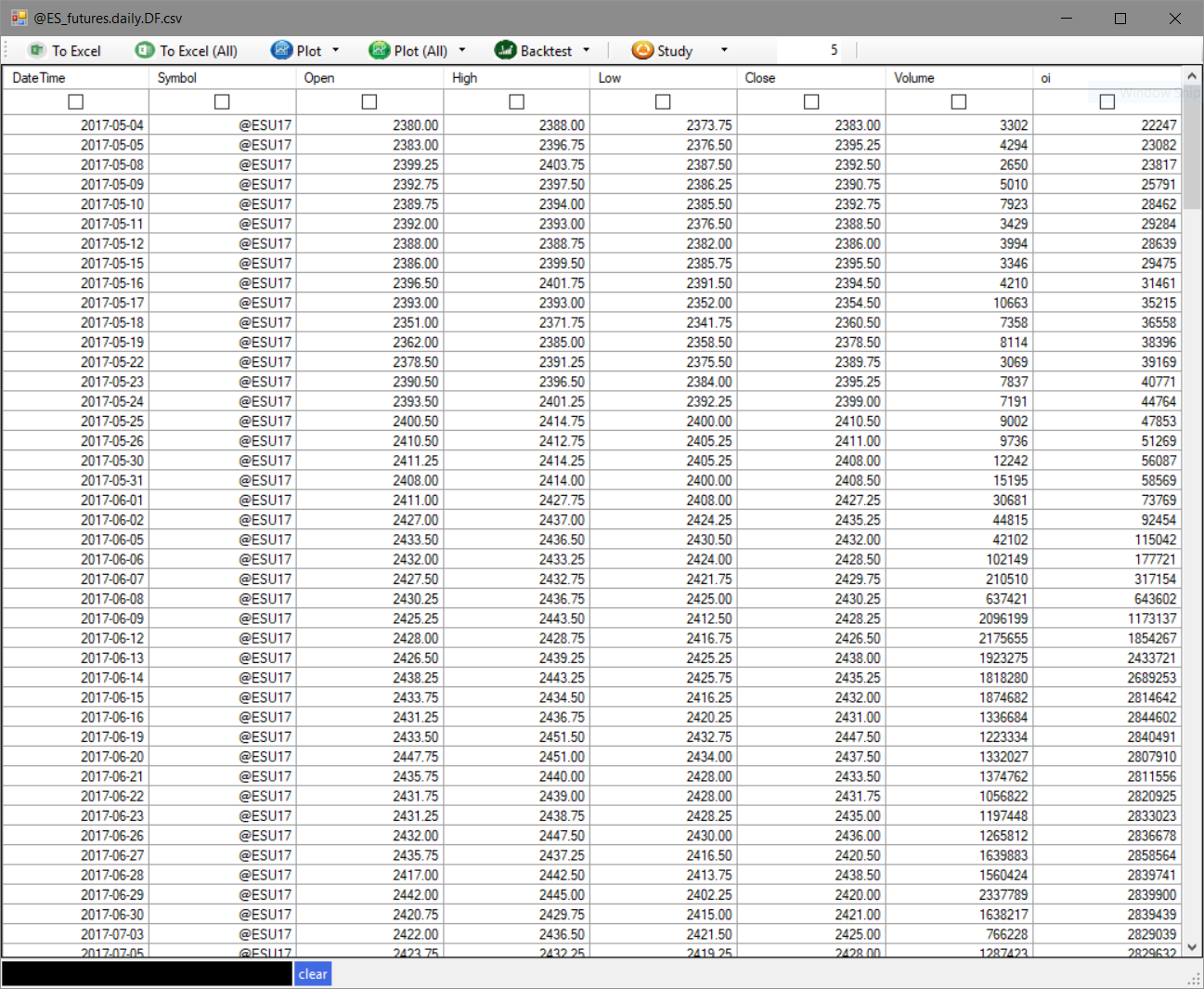
## Main::Data

The tabs (below the button toolbar) allow access to various OCTS functionality. The Data tab displays the historical data files available for analysis, back-testing, etc. All OCTS historical data is organized around “DataFrame” files. DataFrame files are convenient because they can be viewed by any text editor or easily imported into Excel. But more importantly, DataFrame files form the basis for intense data analytics provided by the Python programming language and relevant data science libraries.

To work with a DataFrame file from the Data tab, click on the filename in the “DataFrame Files” list. You can then click the Data Grid button to view the data in table form:



Data from the selected file is displayed in a table layout similar to the following:



In addition to viewing the numeric data, the user can perform the following functions:

1. Export the data to Excel for quick-and-easy analysis (“To Excel” and “To Excel (All)” buttons)
2. Create a plot of one or more columns (“Plot” and “Plot (All)” buttons)
3. Perform a back-test of a specific trading strategy using this historical data (“Backtest” button)
4. Add a “study” to the existing data (such as moving average, exponential moving average, ARIMA, etc.) using the “Study” button

**NOTE:** DataFrame historical data files must be created *by the user*.Other than samples for explanation, no historical data is included with OCTS. (See *Historical Data Form* below for more information.)

## Main::Charts

If instead of viewing/manipulating the numeric data, the user would prefer to see the given data in chart format, click the Chart button on the Main form:



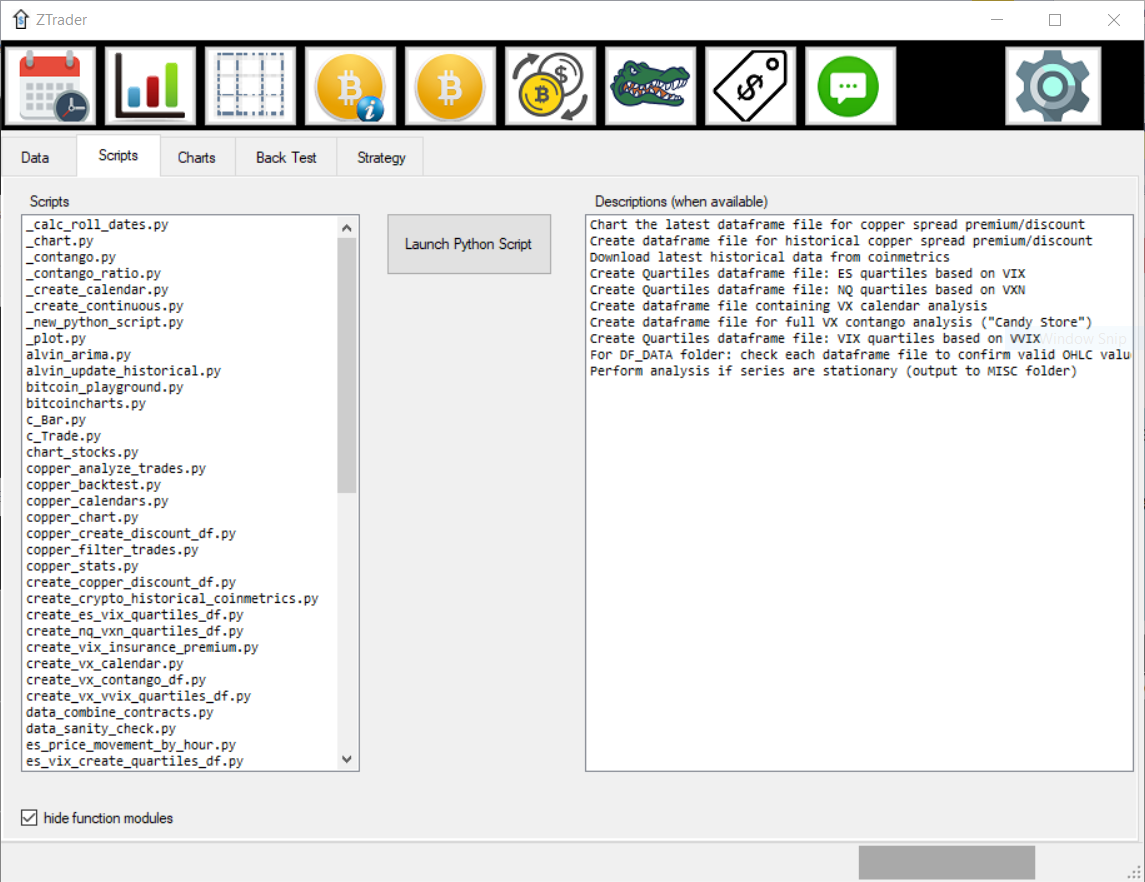
This function will display a chart of the selected historical data file:



The user can display multiple historical data charts as there is no limit to the number of historical charts that can be displayed.

## Main::Scripts

For more advanced data analysis, the OCTS DataFrame file format lends itself perfectly to the Python programming language and its various analytics libraries (Pandas, scikit, numpy, etc.).

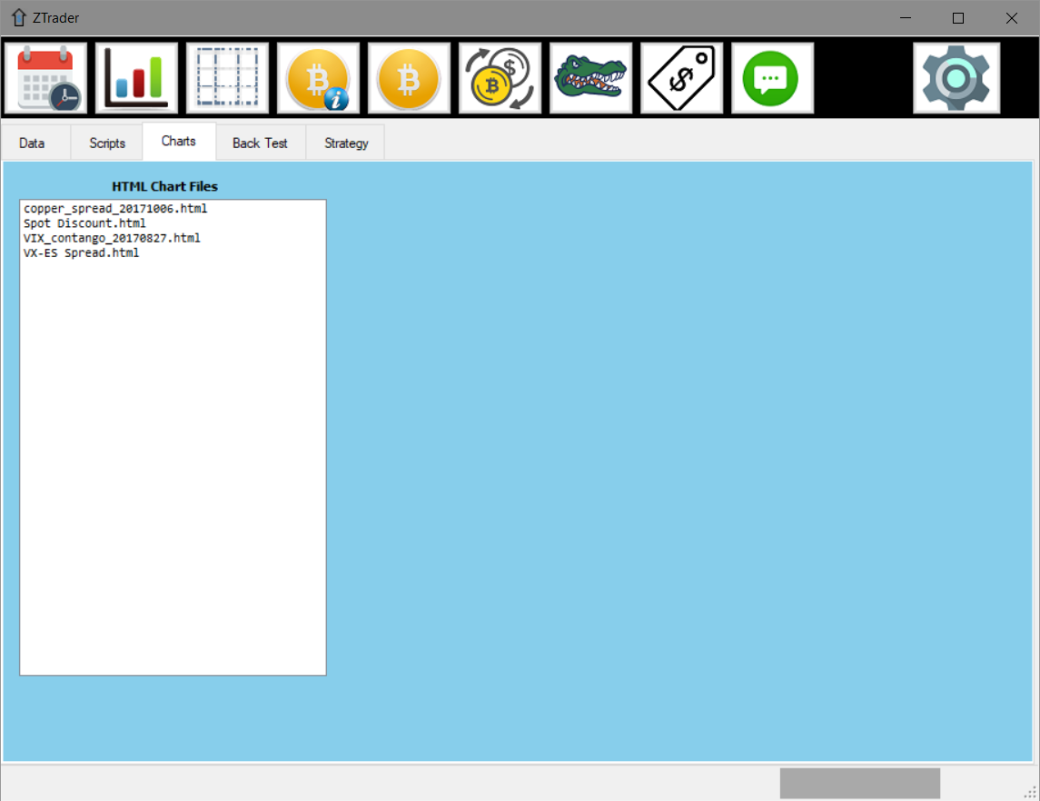


This form shows all the available Python scripts that can be used to perform data analysis. The Python scripts are listed by name in the left window, and a corresponding list of “functional units” exists in the right window. Highlighting a functional unit in the right window will highlight the corresponding Python script in the left window and vice-versa.

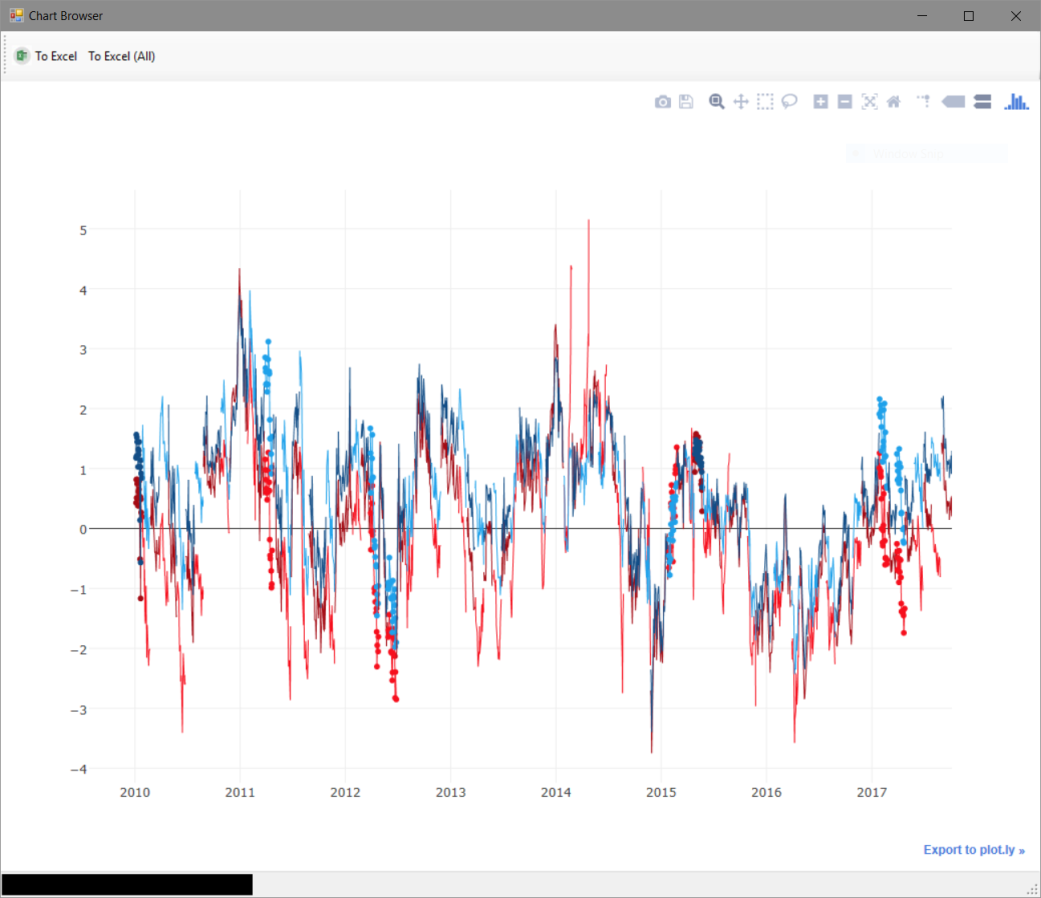
**NOTE:** The data analytics exist to be run on *user-created* Python scripts. Other than samples for explanation, no proprietary data analytics (such as those in the list above) are included with OCTS.

## Main::Charts

The OCTS software is capable of producing a variety of charts and plots. The “Charts” tab of the Main form allows display of a particular subset of charts known as “HTML charts”. These charts are based on the same technology used to create advanced web pages, and they allow for scrolling, zooming, and other advanced chart features:



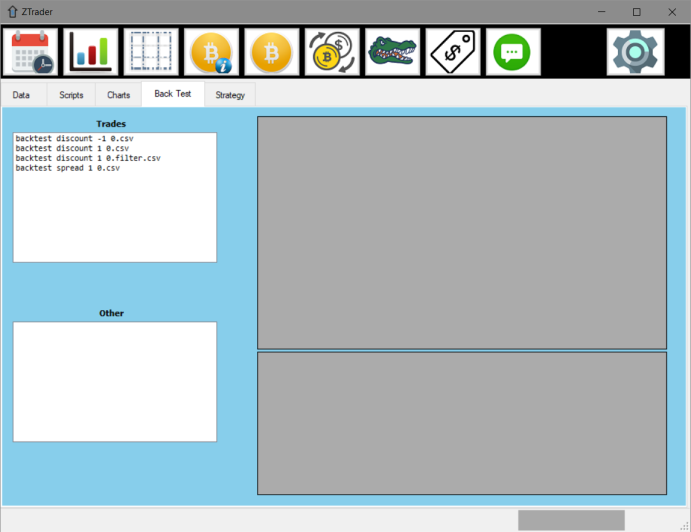
Clicking on a chart in the “HTML Chart Files” list will display that chart in a separate form:



**NOTE:** For accurate display of these “HTML charts,” certain adjustments may be needed to the Windows registry. The Settings form (described below) has the ability to perform these registry adjustments automatically.

## Main::Back Test

Back-testing is the process of taking a trading strategy and running it on historical price data to get an idea of how that strategy would have performed in a given market (during a given time period).

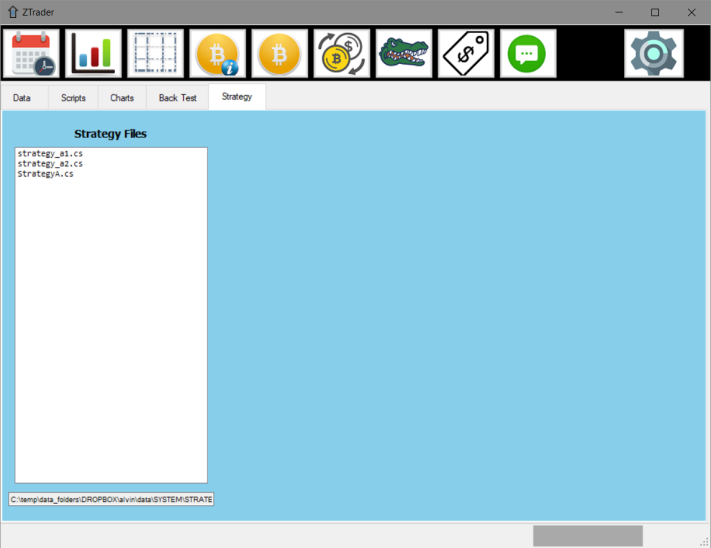


Running a back-test on a specified DataFrame file of historical data produces “Trades” output in a standard .csv (comma-delimited) format. These .csv files can be imported into Excel, examined in a text editor, or otherwise analyzed.

**NOTE:** Back-tests (and strategies described below) exist to be run on *user-created* trading strategies. Other than samples for explanation, no proprietary trading strategies are included with OCTS.

## Main::Strategy

A “strategy” file is a file containing C# programming language code. These files are standard C# code, but they have access to a variety of advanced features via the OCTS programming API:



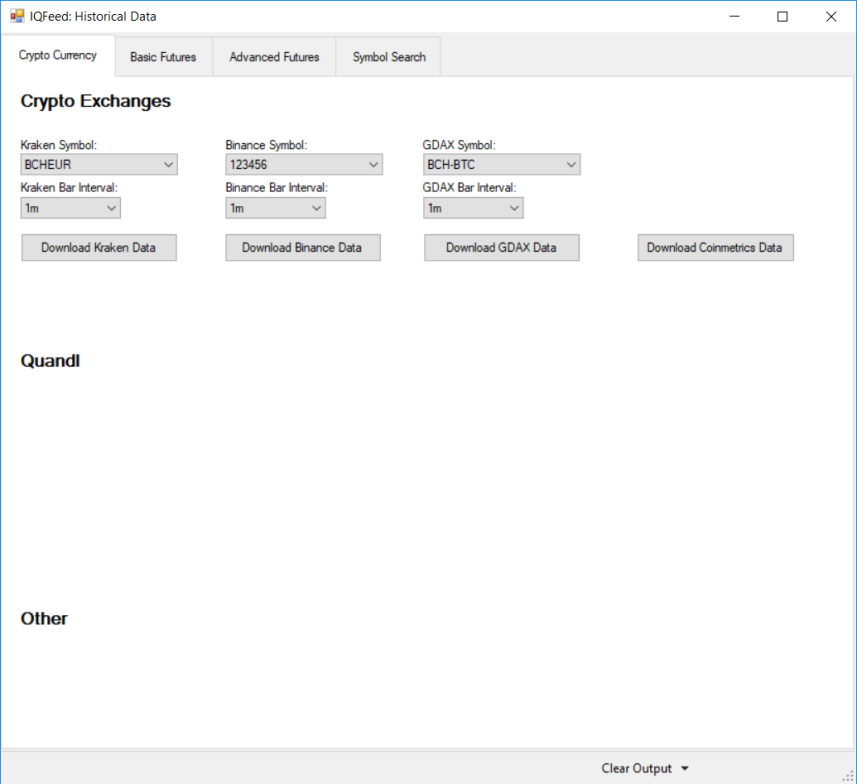
# Historical Data Form

We have seen previously that much of OCTS research and back-testing is organized around DataFrame files. These files typically contain historical price data for individual trading products, but they may contain additional information useful in data analysis.

The Historical Data button on the Main form displays the Historical Data form:



The Historical Data form offers the OCTS user a *huge* selection of traded products for which historical data can be retrieved and formatted into DataFrame files. These DataFrame files can be subsequently used in advanced data analysis, historical charting, back-testing of trading strategies, and other aspects of the OCTS system.



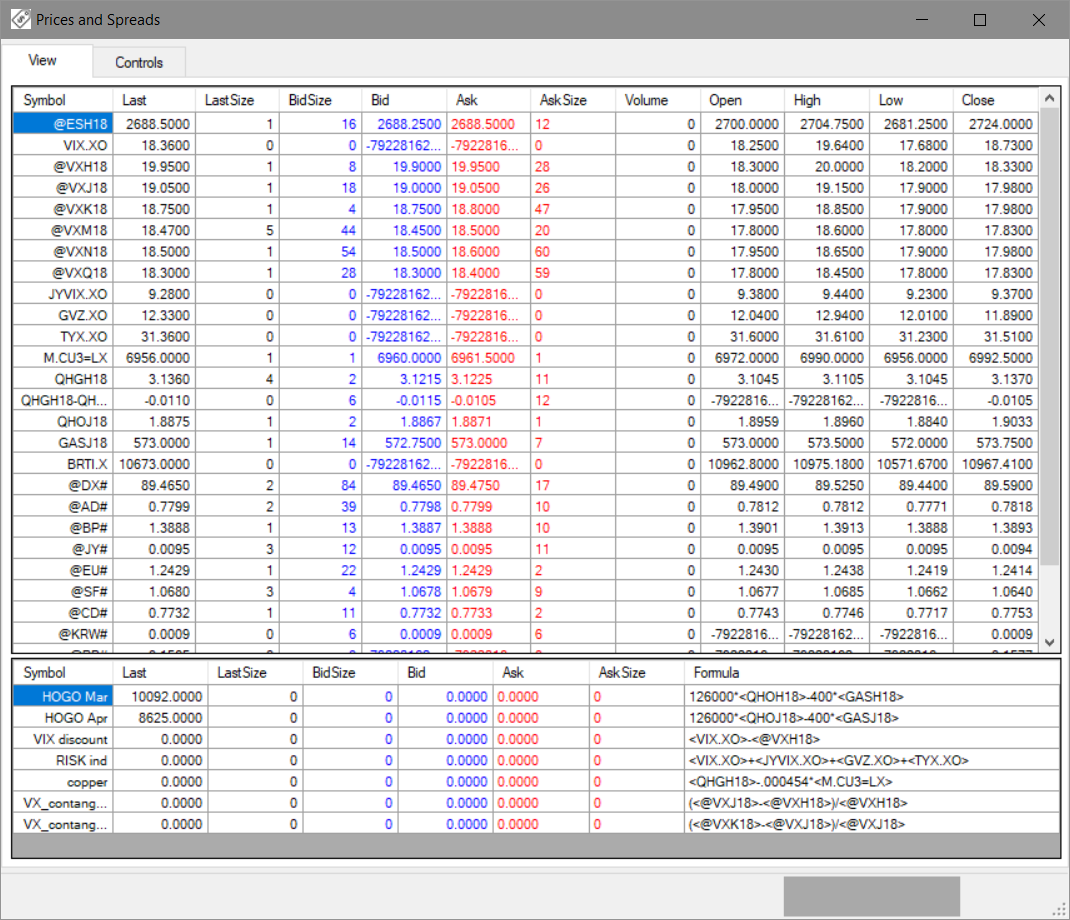
From the Crypto Currency tab in this example, we see that crypto data can be downloaded from Kraken, Binance or GDAX, with other data sources being added frequently.

# Prices and Spreads Form

The OCTS software has its genesis in futures trading, and it still provides a variety of future-related components that can both complement and supplement trading of cryptocurrencies. These can be accessed via the “Prices and Spreads” button on the Main form toolbar:



The Prices and Spreads form displays live futures prices (from CME, CBOE, CBOT, and many other futures exchanges):



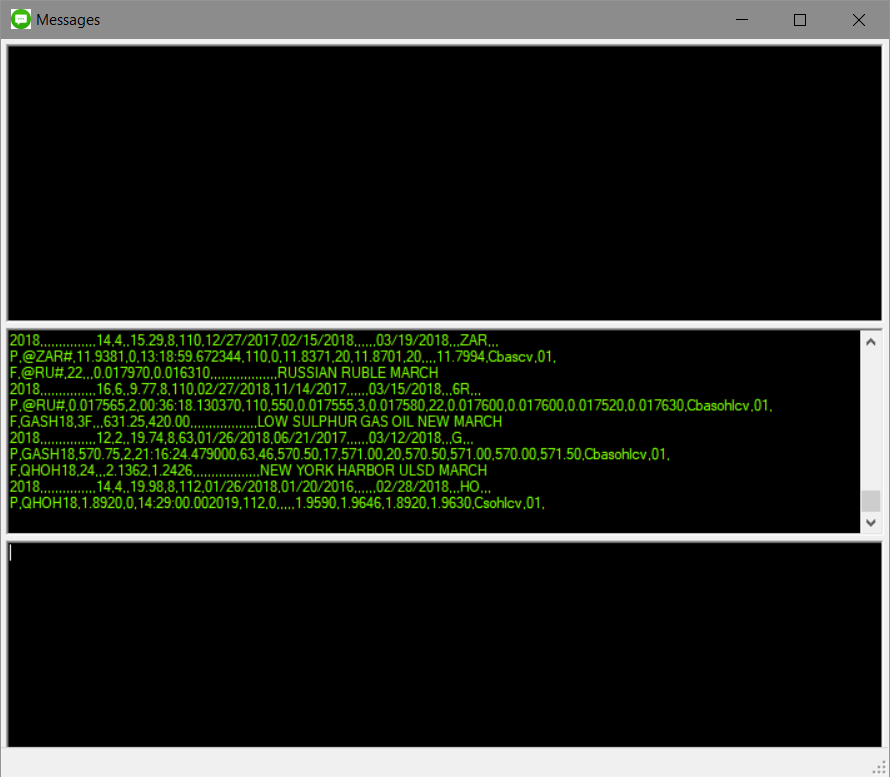
The futures prices are updated in real-time via the DTN IQFeed API interface (separate monthly subscription to DTN IQ data feeds required). The IQFeed API is lightweight, providing insanely fast price updates. Spread prices (bottom pane) are prices calculated based on ratios of two or more futures prices.

# Messages Form

Many functions of OCTS provide customized feedback to the user. There are, however, many general functions that provide simple output, debugging information, or error messages. All of these are viewable from the Messages form:



The Messages form is split into three vertical sections: The top-most section displays standard output. The middle section displays debugging information (typically non-urgent messages regarding OCTS operation). And the bottom section displays any error messages that arise during OCTS operation:

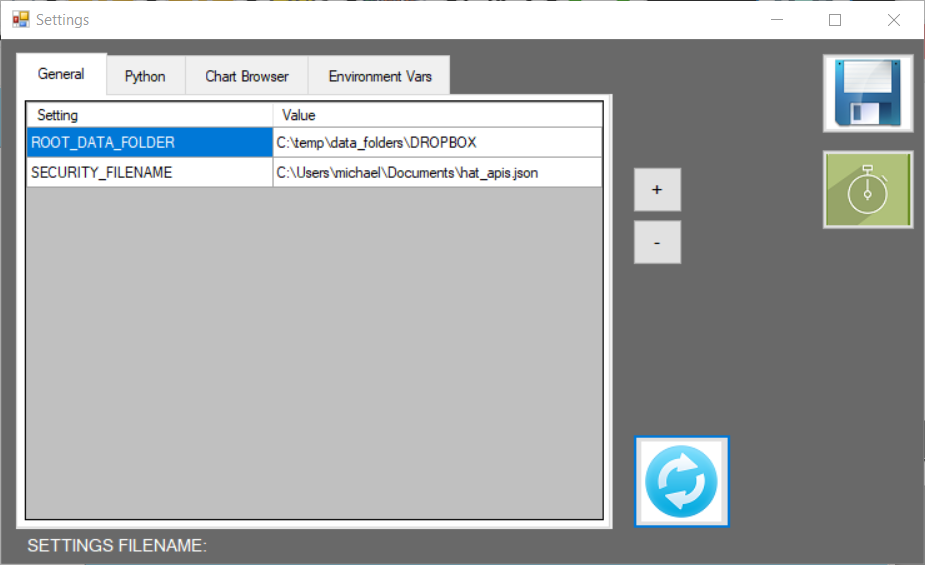


# Settings Form

Like any standard software application, there are a variety of settings that can be tweaked to adjust the performance of OCTS. To change software settings, use the Settings button on the Main form toolbar:



The Settings form has various tabs that divide the software settings into functional areas:

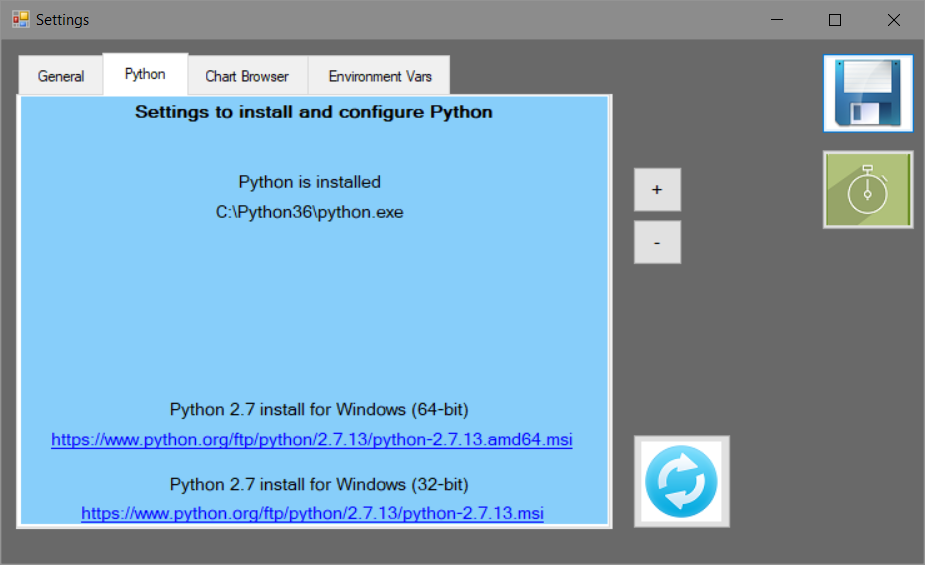


## Settings::General

The General settings tab contains settings that relate to the overall OCTS system. In the example above, settings exist to point to the root data folder and to the security file (for crypto exchange API access).

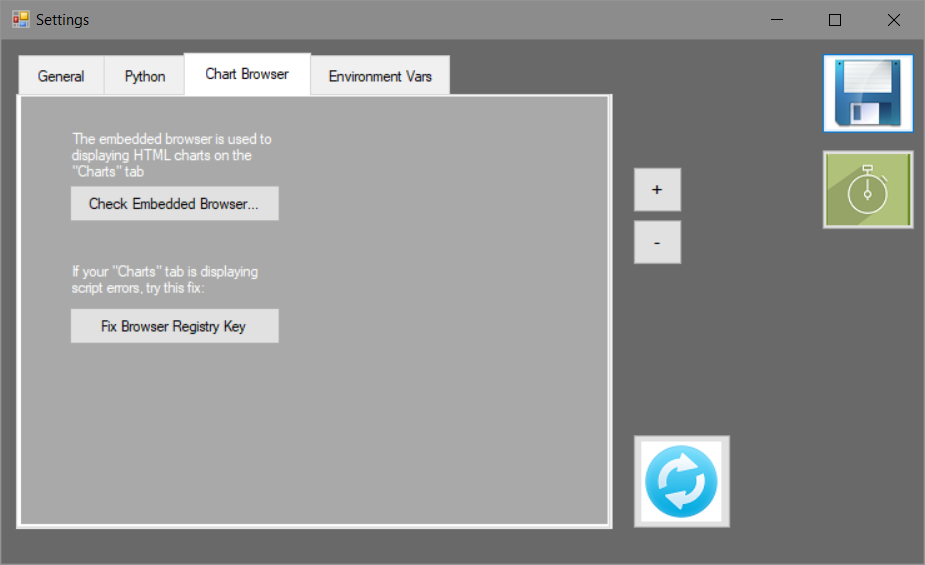
## Settings::Python

As mentioned earlier, Python is the programming language used by OCTS to perform advanced data analysis. The Python settings tab streamlines the process of allowing the user to install the correct version(s) of Python to work with OCTS:



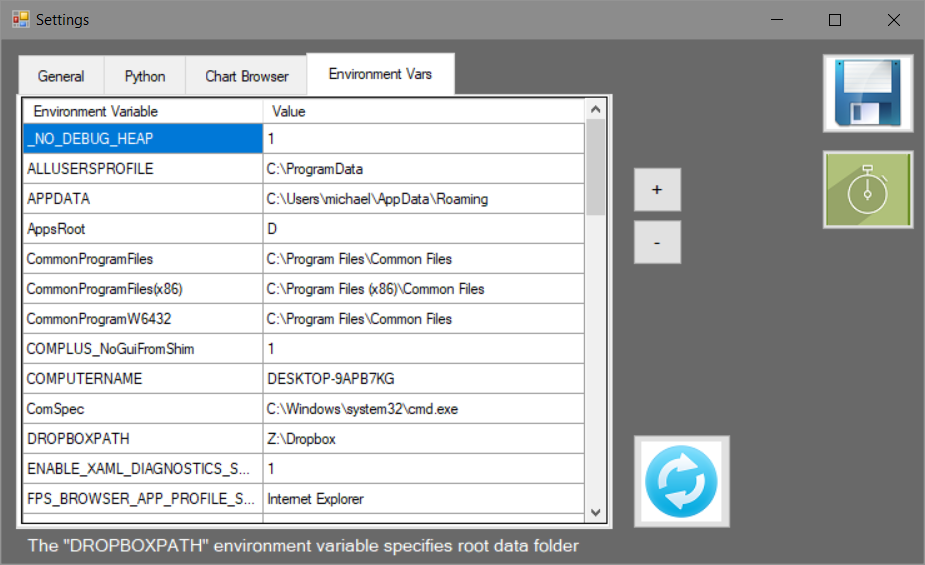
## Settings::Chart Browser

The “HTML” charts mentioned previously (see the Main::Charts section in the above documentation) require certain adjustments to the standard Windows registry settings. The “Chart Browser” settings tab makes it straightforward to perform these registry modifications:



## Settings::Environment Vars

The Windows operating system has various “environment variables” that affect the performance of many Windows software applications. For any Windows environment variables utilized by OCTS, the “Environment Vars” tab of the Settings form provides a shortcut:



# End Notes

Prowl   
<https://www.prowlapp.com/>

Telegram  
<https://telegram.org/>

Python (app downloads and documentation)  
<https://www.python.org/>

Pandas Data Analysis Library (python)  
<http://pandas.pydata.org/>

SciPy Scientific Computing (python)  
<https://www.scipy.org/about.html>

NumPy Scientific Computing Package (python)  
<http://www.numpy.org/>

Binance API  
<https://github.com/binance-exchange/binance-official-api-docs>

Bitfinex API  
<https://docs.bitfinex.com/v1/reference#rest-public-ticker>

Bitstamp API  
<https://www.bitstamp.net/api/>

Bittrex API  
<https://bittrex.com/Home/Api>

Kraken API  
<https://www.kraken.com/help/api>

GDAX API  
<https://docs.gdax.com/#introduction>

Gemini API  
<https://docs.gemini.com/rest-api/>

Poloniex API  
<https://poloniex.com/support/api/>