QuantBench Full Reference:

**Contents:**

Contents

[1 The Interface 2](#_Toc103530342)

[1.1 The Top Pane 3](#_Toc103530343)

[1.2 Price Chart 4](#_Toc103530344)

[1.3 Trade Log 5](#_Toc103530345)

[1.4 Control Panel 6](#_Toc103530346)

[1.5 Console 7](#_Toc103530347)

[2 The Bot API 8](#_Toc103530348)

[3 Quick Use Guide 10](#_Toc103530349)

# 1 Quick Use Guide

Get started with Quantbench

First, load a bot into the application by clicking the Select script button:

Graphical user interface, text, application

Description automatically generated

A file dialog will pop up. Select your bot, in this guide the supplied example bot is used:

Graphical user interface, text, application

Description automatically generated

Once the bot is selected the path will be drawn above the button:

Graphical user interface, text

Description automatically generated

Some bots also output to the console when loaded, the example bot does so:



Next, select a dataset by clicking the Select Dataset button

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Once the dataset is loaded, playback will begin at the default speed and the bot will begin trading:

Graphical user interface

Description automatically generated

The speed of playback can be controlled by dragging the playback speed slider:

A picture containing text

Description automatically generated Text

Description automatically generated

The playback can be stepped through forwards or backwards by using the Step Forwards and Step Backwards buttons, the Step Candles slider controls the number of candles that are stepped by when the buttons are pressed:

Graphical user interface, text, application

Description automatically generated

To move around the price chart, left click and drag the chart

Chart

Description automatically generated

To scale the chart horizontally, use the mouse scroll wheel

A picture containing text, laser

Description automatically generated

To scale the chart vertically, click and drag on the price labels on the y axis:

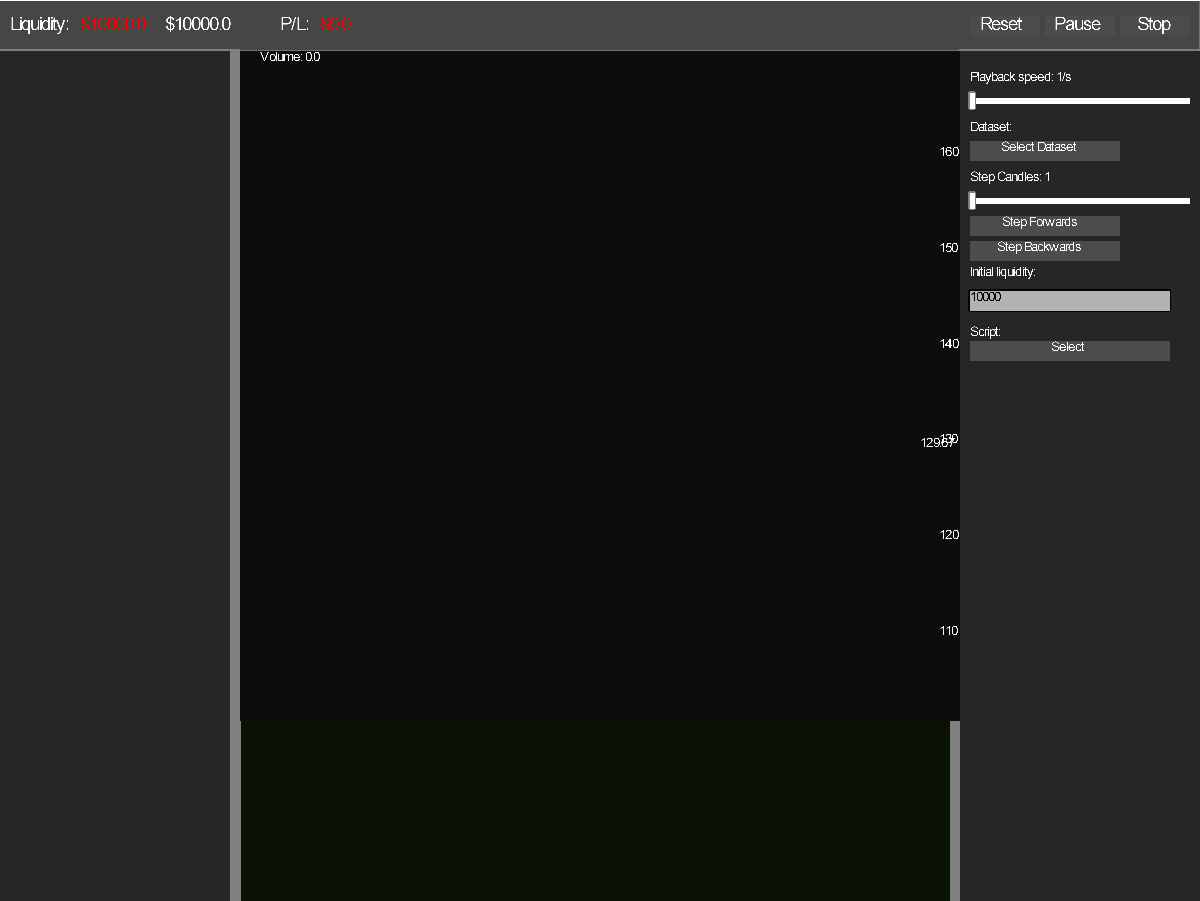
A picture containing chart

Description automatically generated

# 2 The Interface

Upon launching QuantBench you will be greeted with the User Interface. Shown below is an image of the interface with key components labelled. Information on the components can be found below.

**1. Top Pane**



**3. Trade Log**

**4. Control Panel**

**2. Price Chart**

**5. Console**

Shown above is the starting interface with no dataset or bot loaded.

## 1.1 The Top Pane



The Top Pane is made of two main component groups. The first, on the left, is the simulated session account details.

Text

Description automatically generated with medium confidence

The Liquid buying power of the simulated account is shown as Liquidity.

Next to the Liquidity is the effective account Balance, calculated as the sum of the liquidity and value of assets, reduced by the value of any outstanding obligations (Short positions).

The P/L Profit & Loss indicator displays the Profit or Loss during the trading session. It starts at 0 and will increase as the trading algorithm being back tested generates a profit. It decreases when the bot loses money. The P&L is calculated on the Balance rather than the Liquidity, and P&L will be shown on asset holdings as well as liquidity as the price chart playback progresses.



The colour of the Liquidity and P/L indicate whether the trading session is profitable, with green being profitable, and red meaning that the session generated a loss.

The second component group on the top pane are the primary playback controls



The Reset button rewinds the playback to the beginning of the dataset, resets the session balance and liquidity back to the initial value and clears the trade log and console. The bot is also notified that the session has been reset by the ResetSession method being called.

The Pause button suspends the playback and is replaced with a play button that resumes playback. The bot is notified of the state change via the bot Pause and Resume methods.



The Stop button is effectively a combo of the reset and pause buttons. Playback is reset to the beginning of the data set and paused. The session is reset and the bot is notified via the ResetSession method being called.

## 1.2 Price Chart



The price chart displays price action information to the user.

In the top left of the price chart is a volume label, the volume shown is the volume traded at the candlestick the mouse is hovering over.



On the right side of the chart the Y axis is labelled with the trading price, additionally the price at the mouse position is shown.

A picture containing chart

Description automatically generated

On the bottom of the chart the volume traded in each candlestick is shown.

The user can move the view within the chart by clicking on the chart, holding the mouse left button and dragging.

The chart can be horizontally scaled by scrolling with the mouse wheel. To manipulate the chart vertical scale, click on the Y axis labels and drag the mouse up or down to decrease or increase the range.

## 1.3 Trade Log

Trade Log

Image of session trade log

The Trade Log shows the type, quantity and price of trades made by the bot, in order of simulated execution.

The full log can be scrolled through by using the mouse scroll wheel. The most recent trades are shown at the bottom of the trade log, with the oldest at the top.

## 1.4 Control Panel

A screenshot of a computer

Description automatically generated with low confidence

The control panel allows for fine control of the playback. A slider in the range of 0.5 candles to 10 candles per second allows you to control the playback speed to move through the data set faster or slower.

Below the speed slider is the Select Dataset button. Pressing this button opens a standard windows file dialog explorer. Navigate to a data set and select it to load it into the program:

Graphical user interface, application

Description automatically generated

Datasets must be laid out in a specific way to be parsed by the data loader. Columns must not be named, and instead data must be immediate in the CSV file. Each row should be a separate candlestick data point. The Columns of data are listed in the order they are expected:

Open, High, Low, Close, Volume

Example datasets are included in the application distribution to test against and to view the correct data layout.

SPY.csv  
DAX.csv

The Step Size slider allows the skip size to be scaled from 1 to 20 candles. The Step Forwards and Step Backwards buttons when pressed will move the playback forwards by the Step Size Value number of candlesticks.

The Initial Liquidity textbox controls the starting value in the account when a trading session is started. If the Initial Liquidity is not applied to the trading session then resetting the trading session with the Reset button will apply the initial liquidity.

The final button to Select a script will also open a file dialog window, and is used to load up bots written by the user. An example bot has been distributed with the source code in the /ExampleBots/ folder.

## 1.5 Console

A picture containing rectangle

Description automatically generated

The Console displays relevant output from the application and bot. Bots can log to the console to provide debug information when being tested or developed by users.

Newest output is placed at the bottom of the console, with the oldest output at the top. The console content is limited to 500 console messages, and the oldest messages will be deleted when the number of messages exceeds 500. This is necessary to ensure that significant volumes of output do not slow down the performance of Quantbench over time.

# 3 The Bot API

For your bot to be executed by Quantbench it must be implemented with the following characteristics:

The bot script must return a lua table implementing the methods required by the application Bot API. The API methods that must be present are listed in table 1 below:

|  |  |
| --- | --- |
| **Method** | **Usage** |
| Bot:Init(Session) | The Session object is passed to the bot in the arguments of the Init method. The bot should store and read from the Session object to access account liquidity and balance, net position and active price. The Session table is documented in table 2  This method is fired when the bot is loaded by the application. |
| Bot:ClearCandles() | Notifies the bot that playback has been rewound to the beginning of the dataset. The author recommends deleting any data stored about prior playback candles when this method is called.  This method is fired when the Stop or Reset buttons are pressed. |
| Bot:ResetSession() | Notifies the bot that the session has been restarted. The author recommends that any state regarding prior trades should be deleted when this method is called.  This method is fired when the Stop or Reset buttons are pressed. |
| Bot:Pause() | Notifies the bot that playback has been paused. When using timers they should be suspended when the playback is paused.  This method is fired when the Pause button is pressed. |
| Bot:Resume() | Notifies the bot that playback has resumed.  This method is fired when the Play button is pressed. |
| Bot:CandleAdded(Candle) | Notifies the bot that the playback has moved forwards and a candlestick has been added to the price chart. The candlestick has OHLC data and the volume of the candlestick. The author recommends that trading algorithms should be implemented in this method. It is also recommended that candlesticks should be cached in a table by the bot.  This method is fired when a candlestick is added to the price chart.  The order CandleAdded is called in is guaranteed to match the order of candlestick data in the dataset. |
| Bot:RemoveCandles(count) | Notifies the bot that the playback has been stepped back by *count* number of candles. If the bot tracks trades made, then the bot should track the candle id the trade was made at, trades made in the period that is rewound should be invalidated.  This method is fired when the playback is stepped backwards. |
| Bot:MouseHoverCandle(id) | Notifies the bot which candlestick the mouse is hovering over. This number may exceed the current playback position, so it is recommended to clamp the id number to the number of candlesticks that have elapsed.  This method is fired on each frame drawn. |

**Table 1** - The bot API

|  |  |
| --- | --- |
| **Index** | **Description** |
| Liquidity | The amount of liquid money available for the bot to spend on trades. |
| Balance | The total value of liquidity, assets and obligations held in the current trading session. Calculated as Liquidity + NetPosition \* ActivePrice |
| Session.Buy(lotsize) | Buy function that should be used by the bot to notify the application to simulate buying *lotsize* units of the asset  Attempts to buy beyond the Liquidity of the session will fail and output to console. The application will **not** be halted. |
| Session.Sell(lotsize) | Sell function that should be used by the bot to notify the application to simulate selling *lotsize* units of the asset.  Attempts to short sell beyond the Liquidity of the session will fail and output to console. The application will **not** be halted. Short sales are immediately credited to the liquidity, but the liquidity is reduced by any outstanding short obligations (i.e. when a short position is open, the price of the asset is multiplied by the size of the position is considered the size of buy back obligations) |
| Log(string, Color) | Log function that should be used by the bot to output to the console. Different colours of output can be supplied by the bot to assist in adding context or clarity to different desired types of output. |
| NetPosition | The net position of the trading session. A negative value means the position has shorted *abs(NetPosition)* shares, a positive value means the session is holding NetPosition long shares. |
| ActivePrice | The close price of the last candle in the playback |
| Session:AddLabel(str) | Adds a label to the interface control panel. A reference to the label is returned by the function. The text can be modified at run time to indicate debug information or session information to the user. |

It is not recommended to overwrite the functions or values in the Session table. The functionality of Quantbench is not guaranteed should you do so. There are additional values and functions in the Session table that are internal values. Their behaviour is not guaranteed, and they are likely to change.