

BullCharts chart indicator to display text and data

Case study example using BullScript
(based on sample from Des Bleakley)

BullCharts is a powerful and user-friendly charting and analysis system. Anyone who wants to customise chart indicators and dabble “under the covers” with the underlying BullScript code will find it fairly user-friendly, especially if they have some computer programming experience. Even so, it can be challenging because of the inherent power of the BullScript language.

For the people who are new to this subject, and for old hands, this short paper explains some of the details regarding BullScript, and in particular about creating a custom indicator to display text and data on a price chart — as in the sample screen shot in Figure 1.



*Figure 1: A sample string of text and numbers (price-based data)
inserted into a price chart as a "text display" indicator*

Assumptions

In order to create an indicator, and use the BullScript editor to customise an indicator, the user must have BullCharts running, AND also have one price chart displayed for any stock. Then the menu option **Insert > Indicator** is available. This paper assumes that the reader knows how to use the “Indicator Builder” to create or edit the script for an indicator.

For more information and assistance with BullScript, refer to the **BullScript Help** from the BullCharts menu: **Help > BullScript Help**. Refer to the text “Further Reading” on page 4 below for more details.

More information about indicators

BullCharts already has more than 270 indicators. Each one is stored in a separate file on the computer’s hard drive. The BullScript for each indicator can be readily viewed. Any custom indicator that is created is similar to the supplied indicators, except that they are stored in a separate folder on the computer’s hard drive. To find these files, use the BullCharts menu option: **Tools > Preferences > Files (tab)**.

Introduction to “linestyle=text” type indicators.

The BullScript to produce this type of indicator will have three parts, summarised by the three letters: W W W — **What** to put on the chart, **Where** to put this on the chart (ie. at which vertical location, or price value), and **When** to place this across the chart (ie. at what date). This is referred to in the text below.

Robert Brain

Overview

When writing BullScript to produce an indicator, one of the key statements by way of example is of the following form (and this one will produce a line of text on the chart):

```
[Name=Fred; linestyle=text; textalign=above,right;] "HELLO" ;
```

This is actually the “What to put on the chart” first part of the indicator, and BullCharts knows that this is the complete statement because of the semi-colon at the end (which is very important).

To this we can add additional parameters to help customise the appearance as follows:

```
[Name=Fred; linestyle=text; fontsize=12;
                                textalign=above,right;] "HELLO";
```

where:

Name= Indicates the name for the element that is about to be plotted on the price chart. In this case **Fred**

linestyle=text This is the key parameter that specifies that this “indicator” will be a line of text, rather than a regular indicator. Instead of “**text**”, other possible values include: **line**, **dotted**, **stepped**, etc.

fontsize= This is one of several parameters that can be specified to customise the appearance of the indicator. Refer to the BullScript Help for other possible parameters.

textalign= This is used to specify the location of the text. In this example, “**textalign=above,right;**” will place the text “**above**” the price action, and it will be right-justified (ie. the right hand end of the text string will be at the location that will be specified later).

Note that this line of text has a square bracket to start, and one to end, and that the colours of the text shown in this sample are exactly how BullCharts will display this script text in the BullScript editor.

The above line “**..[Name= ... right;]**” describes the features of WHAT is to be displayed, and is then followed by two items which explain WHEN across the chart to display it, and WHERE abouts on the chart (vertically) it is to be displayed. This will make more sense with the following description.

So let’s look at the following complete example which incorporates the statement above, followed by another important element:

```
[Name=Fred; linestyle=text; fontsize=12;
                                textalign=above,right;] "HELLO"; CLOSE;
```

We described “**textalign=above,right;**” above.

The portion “**HELLO**”; says that the text string **HELLO** is to be placed on the chart, and the portion **CLOSE**; says that the text string is to be placed relative to the share price Close value.

Note that the BullCharts Formulae **CLOSE** is actually a one-dimensional array that contains all of the Close values for the stock whose chart is displayed. So the sample script shown here will actually place the text “**HELLO**” multiple times across the chart — once for each Close price shown on the chart (as shown in the screen shot in Figure 2 at right).

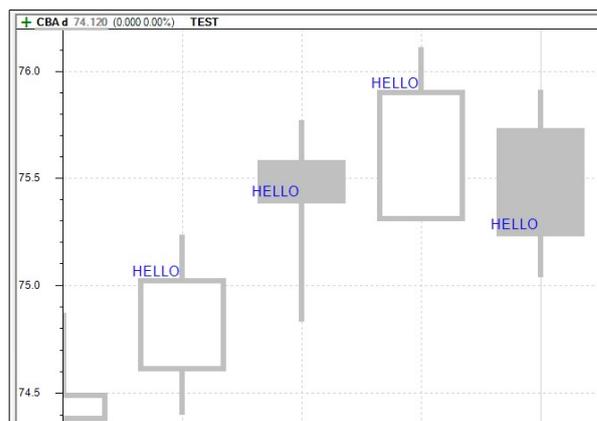


Figure 2: The word *HELLO* displayed adjacent to every Close value

Note that the `CLOSE`; is the part of the indicator: **Where** to put this on the chart (ie. at which vertical location, or price value).

Refinement

In the example shown above, and displayed in Figure 2 above, the word `HELLO` is shown on every candle (every Close value) across the chart. But if we want to have it displayed only on the last candle, then we need to modify the above text as follows:

Replace the script: `"HELLO"; CLOSE`;

with: `IF(Lastbar, "HELLO",undefined); CLOSE`;

where each portion of this is explained as follows:

`IF(` — Is the start of the script "IF" statement, which usually has three values within the pair of parentheses.

`Lastbar` — Is the first parameter of three parameters in the IF statement, and in this case is a Logical Expression that will have the value "True", or "False".

The value for `Lastbar` is a user-defined script variable and is calculated elsewhere within the script.

It is actually produced with the following script statement:

```
Lastbar := BARNUMBER = LASTVALUE(BARNUMBER) ;
```

`"HELLO"` — Is the string of text to be displayed.

`undefined` — This BullCharts keyword is used here to specify that if the value of `Lastbar` is false, then nothing is to be displayed. See the next paragraph for an explanation for this.

`CLOSE`; — This states that the text string is to be placed vertically (along the y-axis) at the position relative to the share price Close value.

How to interpret the IF statement:

If the first value inside the parentheses has the value "True", then the value for `strategy` is displayed on the price chart.

If the first value inside the parentheses has the value "False", then the BullCharts keyword `undefined` says to display nothing.

The effect of all this is that the calculation for `Lastbar` will be evaluated to be true only for the last bar (ie. candle) on the chart. So the text will only appear adjacent to the very last bar (candle).

Sample Script

The sample BullScript shown on the following pages is a real example, with relevant comments included within the script. This sample will produce the output as shown in Figure 1 on page 1 above.

In order to assist the reader with the sample script code on the following page, we have tried to use the following convention to help distinguish between our own Variables, and the BullCharts internal FUNCTIONS and FORMULAE. Where we have created a Variable, to store a value, we have used an initial Capital letter and black text (eg. `Printposn`). To help identify the internal **FUNCTIONS** that are used in BullCharts, we have used all capital letters and a dark blue text (eg. `HHV(HIGH,50)`). Even so, there might be some instances that are not clear.

Further reading: For more information on this topic, refer to the BullCharts Script Help as follows. Select BullCharts menu item: **Help > BullScript Help**, then click on the “Search” tab, and in the search field box type the text: **displaying text**, then in the list of topics that are shown for the search, double click on the “Displaying Text” entry.

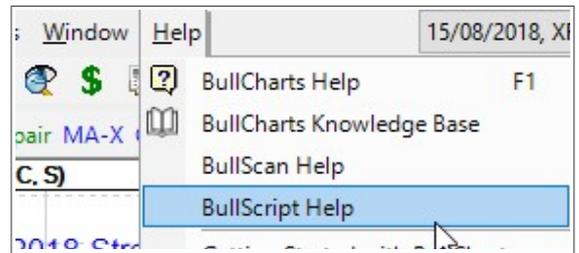


Figure 3: The menu option for: BullScript Help

BullScript case study sample — based on a portion of BullScript from Des Bleakley (compiled by RB)

[**description** = This indicator script is an extract from one of Des's Custom Indicator BUY/SELL strategies (with many lines removed for simplicity in this example).

Summary line gives :

Brief description of strategy

P - Profit - compounded over selected period - default period 65 bars - user selectable

WR - Win Ratio - percentage of wins from total trades

MDD - Maximum Draw Down %

TIM - Time In Market %

PPA - Profit Per Annum % (Profit / num years)

Traders Rate of Change - TROC

ADVICE - BUY, SELL, HOLD, STAY OUT determined by strategy. ;

author=Bleakley, Des; target=price;]

{ **First, prompt for some input parameters** }

Printposn := input("Print Position ", 5, 1, 20); { Specifies the vertical "print" position }

History_bars := input("History bars ", 65, 0, 1000); { The number of bars to look back }

Expr := expression("Expression");

Method := inputma("Method", SIMPLE);

Lastbar := barnumber=lastvalue(barnumber);

PrintpositionH := HHV(HIGH, 50) * (Printposn/10+1);

Strategy_num := 6 ;

(5); 65 bars(y) from 15/05/2018 Strategy Number:6 | P:% | WR:% | TROC:

[color = blue];

{ **** Long Bar Summary = LBS **** }

[Name=LBS; { The name for this element (indicator) that will be plotted.}

linestyle=text; { The "type" of indicator that will be plotted. In this case a "text" indicator.}

fontsize=12;

fontstyle=bold;

textalign=above,right; { Specifies to "align" the text "above" the set location, and "right" justified.}

visible=false;] { Specifies to "hide" this text (the converse is "true". }

{ ***** Firstly, assemble the line of text to display ***** }

{ **This displays the Strategy details for this example** }

Strategy := { The name of the script variable "strategy", which will be a string of concatenated text. }

"(" { The pair of double quotes encloses some text to display.}

+ { The plus sign will concatenate the previous text item with the next one.}

```

Printposn      { Display the value of the variable Printposn. }
+              { The plus sign will concatenate the previous text item with the next one.}
") - "        { The pair of double quotes encloses some text to display.}

+              { The plus sign will concatenate the previous text item with the next one.}
History_bars  { Display the value of the variable History_bars.}
+

" bars("      { The pair of double quotes encloses some text to display.}
+
{ RTY }       { If the parentheses were omitted here, it will display the value of variable RTY. }
+

"y) from "    { The pair of double quotes encloses some text to display.}
+
HIST( NOW, History_bars) ;
{ The semi-colon is the instruction to end the above assembling of text. }

{ ** Now to specify that the text/data string is only to be displayed at the "lastbar" and not every bar }
{   The IF statement is evaluated for every individual bar/candle across the displayed time line, and }
{     if the actual "lastbar" has been reached, then display the text string in "strategy" }
{     concatenated with any other text and data that we want displayed. }

if ( Lastbar, { IF the value of the variable "Lastbar" is TRUE, then display the next bit...}
Strategy
+
" Strategy Number:"
+
Strategy_num
+
" | P:"
+
{ round(final_profit) }
+
"% | WR:"
+
"% | TROC: " ,undefined); { ...display the text string, otherwise, display nothing. }
+

PrintpositionH ; { This specifies the Height up the price scale to position the text. }

```



(5); 65 bars(y) from 15/05/2018 Strategy Number:6 | P:% | WR:% | TROC: