A BullScript Primer

To help BullCharts users understand and learn about
the BullScript language, functions, keywords, etc.

Document status — Fourth draft

19 August 2016

This document is an early draft of an attempt to pull together useful and relevant information about BullCharts and BullScript, to assist newcomers to BullScript to understand more about BullScript, and how to use it. There are other documents that should also be referred to.

Introduction

For a computer programmer who wants to utilise BullScript, there are some challenges; but for someone not experienced with computer programming the challenges are a lot greater.

What is BullScript?

BullScript is the computer language that is used behind the scenes within the BullCharts charting software. All chart indicators and scans rely on BullScript “under the covers” to actually function. It is also utilised in the TradeSim back-testing system.

Most BullCharts users never get to see the BullScript language in use, because most of what they need to do can be achieved with the user-friendly menus and drop-down options to make choices, and select parameters and options.

However, there are times when some users might want to do something a little out of the ordinary — like tweak an indicator, and create a special formula to obtain a trading edge. This is where access to the BullScript language can be beneficial.

This BullScript Primer

This particular document is structured into two parts:

- Part 1 includes a few pages that show an example of BullScript in action, with explanations for key aspects. This will help the novices to get a basic understanding of how it works, and how to interpret the scripting computer language.
- Part 2 contains a series of tables, each of which provide a list of different types of BullScript elements. It can be used as a reference resource to help find, and understand, information for key aspects of elements of the BullScript language.

More BullScript Help

There are several ways to pick up more information about BullScript, including:

- The BullScript users manual.
- The online BullScript Help.

After picking up some basic knowledge of BullScript, one good way to try to use and learn more about it is to utilise the online BullScript Help facility within BullCharts. To do this, access the script editor as follows:

1. From the BullCharts menu: Help > ...
   (either: BullCharts Help, or BullScan Help, or BullScript Help).
2. For help with editing a BullScript scan, use the Script Helper, and either press Function key F1, or click on the “Helper” button, then the “Help” button.
3. For help with a chart indicator, access the Indicator Builder, then with a new indicator or when editing an existing indicator, either press Function key F1, or click on the “Help” button.

In addition to these resources, make sure to study all the sections of the BullScript manual, entitled “Getting Started with BullScript”. It has very useful sections on the following aspects:

- Learning BullScript and a Language Overview.
- Using the ‘If’ statement.
- The Hist, Future and Previous Functions.
- Variables (ie. what are they?).
- Defining Formulae with Parameters
- Generating Multiple Plots.
- Displaying Text.
- More about Attributes (and specific attributes like: Color, FillStyle, Target, etc.).
- Best Practices.
- Several Sample Scripts.
- Several Sample Indicators.
- Troubleshooting and Error Codes.
- Index (at the very back).

Robert Brain
(User Group convenor)

Table of Contents

Introduction....1
What is BullScript?....1
This BullScript Primer....1
More BullScript Help....1
PART 1 — BullScript for beginners....3
Introduction....3
BullScript preamble....3
Sample BullScript — The Moving Average indicator....4
PART 2 — Reference information....8
Table 1 – Introduction, Basics, Terminology....8
Table 2 – Price Variables (O, H, L, C, etc.)....10
Table 3 – Security Variables (aka fundamental values)....10
Table 4 – Attributes....11
Table 5 – Functions — introduction....13
Table 6 – Functions — general....13
Table 7 – Functions — mathematical....14
Table 8 – Functions — candle patterns....15
Table 9 – Functions — pattern recognition....16
Table 10 – Functions — date and time....16
Table 11 – Functions — highest/lowest....17
Table 12 – Functions — indicator....17
Table 13 – Functions — text....18
Table 14 – Functions — TradeSim....18

PART 1 — BullScript for beginners

The next five pages of information started life as a standalone document to assist BullCharts users with their early use of BullScript. It is reproduced here with enhancements to demonstrate one real use for BullScript, and how to understand some of the key elements and aspects of BullScript.

Introduction

BullScript is already used in the BullCharts software to define chart indicators, and BullCharts scans. For most users most of the time, there is no need to look at the underlying script because the menu options and the drop-down selections enable users to adjust the chart indicator parameters and the scan criteria details for the majority of requirements. However, there can be times when a BullCharts user might want to tweak the underlying BullScript code, or perhaps even write their own code.

In the example shown and explained below, we look at using BullScript in a chart indicator, and in particular for the Moving Average chart indicator.

BullScript preamble

Following is some introductory material on the subject.

Three price variables:
`high`
`low`
`close`
 	
Note: These price variables can be abbreviated to just the first letter, and they can be upper or lower case:
`H`
`L`
`C`
	These three ***price variables*** are samples of how BullCharts handles the security price data (there are others — open, trades, volume, value).
	The variable `high` represents the high price for all bars across the price chart. With reference to Figure 1 below, understand that each high price on the chart is stored within the ***price variable*** called `high`. That is, it is a single-dimension matrix containing a list of all the price high values. In fact, for the period shown on the chart here from 10 to 25 January, the high values are: $37.72, $37.20, $36.64, $36.7, $36.4, $36.65, $36.72, $36.54, $36.85, $37.15, $37.22 and $37.27.

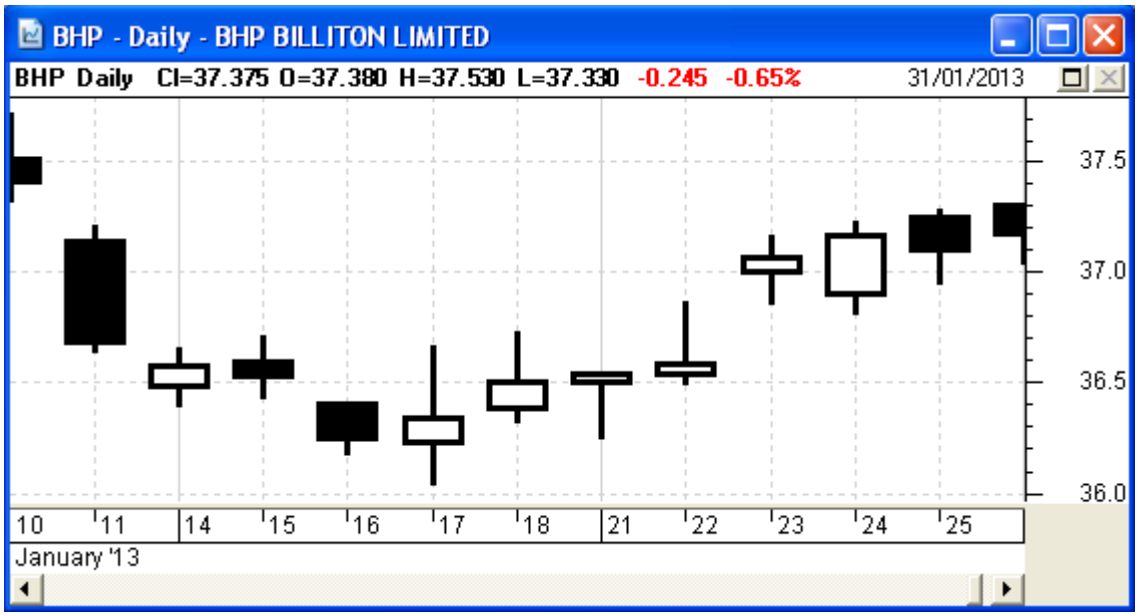

Figure 1: Daily candlestick price chart for BHP.

Variables and Formulae — sample:
 	
`TypicalPrice :=`
`(high + low + close)/3;`
	The two characters "**:=**" are known as the ***assignment operator***. In a statement like the sample at left (which might normally be shown across one single line, and not split across two as shown here), this operator has a function or calculation on the right-hand side, and it specifies that the result of the function or calculation is to be stored in the ***variable*** named on the left hand side (a typical computer programming standard).
	In this example **`TypicalPrice`** is a variable name which will store the answer to the calculation from the right-hand side. This calculation is simply the addition of three price variables, with the result divided by three (to simply give the average of the three values). Understand that because each of the three price variables actually contains a list of values, then the resulting answer will be a list of values.

Plotting a basic Moving Average
 	
`MA(C,28,SIMPLE);`
 	
Could also be written as:
`ma(Close,28,Simple);`
	The simple statement shown at left is the Moving Average indicator function (one of many indicator functions), and when appearing in BullScript like this, followed by the semi-colon character, it will cause the Moving Average indicator to be drawn on the price chart. The elements of this sample statement are:
	MA — specifies the function (the moving average indicator function)
	C — based on the Close price,
	28 — a 28 period Moving Average,
	SIMPLE — a simple Moving Average, as opposed to Exponential, Weighted, etc.

Sample BullScript — The Moving Average indicator

line no.* The following includes key lines of BullScript code for the Moving Average indicator. Each element of the script is explained in the next table below. (The {,,,} lines of comments have been added here to assist the reader.)

```
 1  {Anything inside curly braces is a comment}
    {The first few lines of a script often include useful
    informational text}
 2  [description="Average of price (or other value) calculated
    over a period of time..."]
 3  [target=Price; category=Moving Average]
 4
 4  {We are leaving a blank line above just to aid readability}
 4  {The next few lines are setting up the calculation
 4  and getting the parameters that are needed}
 5  expr := expression("Expression"); {eg. Close, Open, etc.}
 6  method := inputma("Method",SIMPLE); {eg. simple, exp, etc.}
 7  n := input("Time periods",14,1); {time periods}
    {Note that a comment can be included at the end of a line}

    {The next line is the calculation of the MA}
 8  res := ma(expr,n,method);

    {The next line generates the plot of the MA}
 9  res;
```

* – Note that the "line number" in the first column of this table is only included here to help the reader refer to specific lines if script. They do not appear in the real BullScript routines.

Statement from the script	Description and comment
Line 1 `{Anything inside curly braces is a comment}`	This line simply says that anything that is inside a pair of curly braces { } is regarded as a comment. It is only for the information of the user. The BullCharts system ignores it. Serious computer programmers already understand the importance of including comments inside a computer program, or in a script routine, because it is regarded as internal documentation to help the reader to understand what the lines of script are attempting to do.
Line 2 `[description="Average of price (or other value) calculated over a period of time..."]`	The pair of square brackets [and] indicates one or more ***attributes***. In this case the attribute called **`description`**. The text that follows within the quotation marks can be referred to elsewhere within the script routine.

Line 3
 	
`[target=Price; category=Moving Average]`
 	
Line 2 and 3 could be combined into one script statement:
 	
`[description="Average of price (or other value) calculated over a period of time..."; target=Price; category=Moving Average]`
	The pair of square brackets indicates one (or more) ***attributes***. In this case, the two attributes: **`target`**, and **`category`**, which both have special meaning.
	The **`target`** attribute specifies where the MA will be drawn — in this case on the **`Price`** plot (as opposed to the other possible targets of: **`Volume`**, **`New Pane`**, etc.).
	The **`category`** attribute literally specifies the "category" for this item. For the MA, this means that when you use the BullCharts menu item ***Insert > Indicator*** (see Figure 2 below), to insert an indicator, the dialogue drop down labelled "Category" will show the MA within the list of "Moving Average" items. That is, you can create your own indicator **`category.`**

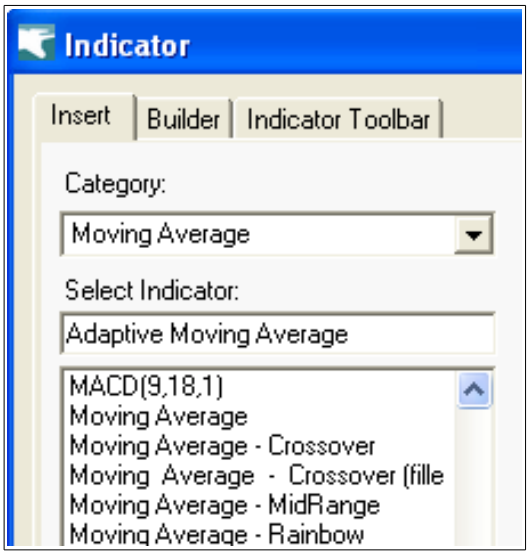

Figure 2: The "Insert Indicator" dialogue box — Category selection.

Statement from the script	Description and comment
Lines numbered 4: `{We are leaving a blank...}`	These lines have a blank line to start with. This is a good practise to make it easier to read and understand the script later. And then there are more comment lines.
Line 5 `expr := expression("Expression");`	The BullCharts menu ***Insert > Indicator*** dialogue (as shown in Figure 3 below) prompts for three things:- "Expression", "Method", and "Time Periods". The statement at left prompts for a value for **"Expression"**, and stores the entered value in a variable named **`expr`**. This will be one of:- Open, High, Low, Close, Volume, Value or Trades (these are known in BullCharts as ***price variables***).
alternative code: `price-parameter := expression("Select the price parameter")`	The Line 5 above could just as easily have been something like that shown at left (where **`price-parameter`** is the name of the variable that will store the answer).
`{eg. Close, Open, etc.}`	On the end of Line 5 we have included this text. Because it is enclosed inside curly braces { } it is treated as a comment. We are including it here as internal documentation to help the reader.

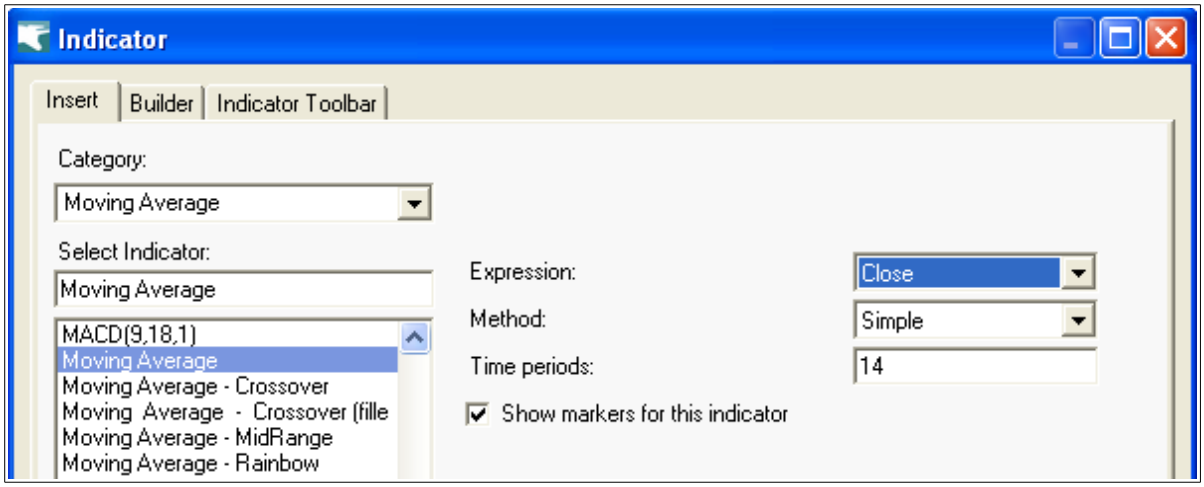

Figure 3: The "Insert Indicator" dialogue box to enter parameters.

Statement from the script	Description and comment
Line 6 `method := inputma("Method",SIMPLE);`	The keyword here **`inputma`** specifically prompts for the type of moving average to be entered (ie. simple, exponential, etc.). The word "Method" is displayed in the prompt (see the Figure above), and **`SIMPLE`** is the default value that is offered (note that the text included here as the default must be a valid item from the list of available items). The actual value that is entered or selected is stored for later use in the variable named **`method`**.

Statement from the script	Description and comment
Line 7 `n := input("Time periods",14,1);`	The keyword here **`input`** prompts for a numeric value to be entered. The text “Time periods” is displayed in the prompt (see the Figure above), and the two numbers here are: • 14 — this is the default value offered, and • 1 — this is the minimum value. The actual value that is entered or selected is stored for later use in the variable named **`n`**.
Line 8 `res := ma(expr,n,method);`	The text: **`"ma(expr,n,method)"`** is the moving average indicator function, and specifies that the Moving Average values are to be calculated (for the current security), using the values stored in the three variables that were input above:- **`expr,n,method`**. The resulting moving average values are stored in a variable named “**`res`**”. (This is simply an abbreviation for the word **`result`** .)
Line 9 `res;`	This simple statement has the name of a variable, in this case **`res,`** followed by a semi-colon. This causes whatever has been specified in the variable **`res`** to be drawn on the price chart.

This is the end of the example and discussion in Part 1. The following material in Part 2 is for reference.

PART 2 — Reference information

Table 1 – Introduction, Basics, Terminology		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`{ }`	comments	Curly braces are used to enclose comments.
`[ ]`	attributes	Square brackets are used to enclose *Attributes*. eg. **`[color=green],`** or if more than one attribute, separate with a semi-colon eg.**`[color=red; target=price]`**
`:=`	*Assignment operator*	The information on the right side of this operator is a calculation of some form. The information on the left side is a variable or function name where the answer will be stored.
`;`	Semi-colon	(1) Used to separate ***attributes*** (see ***attributes*** details). (2) Used to separate the calculations that are to be plotted so that more than one plot can be generated.
`Expression` **`expr`**		The term ***expression*** (and it's abbreviation ***expr***) is widely used herein, and in fact is to do with script programming in general. It refers to a combination of *keywords, operators*, *variables*, and *constants* that result in a *text string*, *number*, or other *object*. An expression can perform a calculation, or manipulate text characters, or test data.
`comments`		Comments in script are enclosed in curly braces: { }
`Security Variables`	*Security Variable*	To reference a ***security variable*** in a calculation, enter the word code **`Security.`** (including the dot) followed by the variable to be included. For example **`Security.PERatio`** . See the list of ***Security Variables*** in Table 2 below.
`Function`		BullCharts includes many *Functions*, which are used to perform a calculation and derive a value (or a series of values). The *Functions* can take one or more input values to be used within the calculation. There are several types of *Functions*. Some examples are listed further down in this document.
`Formula`		The word *Formula* is often used as another name for *Function*. The key difference is that a *Function*, is already supplied whereas a *Formula* is defined by the user.

Table 1 – Introduction, Basics, Terminology		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
Numeric Data	*Data Type*	Most of the BullScript functions and calculations work with numerical data. BullCharts stores data values with several significant digits, and can be displayed with or without decimal places.
Text Data	*Data Type*	Text information can be processed, and is often referred to as a character string.
Boolean (Yes/No) Data	*Data Type*	In computer speak, “Boolean data” generally refers to a condition that is either True, or it is False. For example “Has the price crossed above it's 7 day moving average?”. The condition “False” can be represented numerically by the value zero, while the condition “True” can be represented by the digit for one (or in BullCharts can be any positive number).
Date Data	*Data Type*	Dates can be stored and manipulated, and used in calculations. A date is stored in one number, and stores a value for each of: year, month, day, hour, minute and second. Each of these elements can be extracted for use.
Enumerations	*Data Type*	An ***enumeration*** is a set of values that are defined for internal use. For example, BullCharts supports one enumeration for the Moving Average which contains the values: simple, exponential, weighted, etc.
Undefined	*Data Type*	This refers to the result of a calculation, or price data, where the answer is not meaningful. For example, for a 10-day Moving Average, the first nine days will not have a meaningful value for the MA.

Table 2 – Price Variables (O, H, L, C, etc.) (ie. open, high, low, close, volume, value, trades)		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
Open	Price Variable (abbr O)	The opening price, or first traded price.
High	Price Variable (abbr H)	The highest price achieved during the interval.
Low	Price Variable (abbr L)	The lowest price achieved during the interval.
Close	Price Variable (abbr C)	The closing price, or last price traded for the interval.
Volume	Price Variable (abbr V)	The total volume traded in the interval.
Value	Price Variable	The total cost of all trades in the interval. Also called turnover.
Trades	Price Variable	The number of trades made in the interval.

Table 3 – Security Variables (aka fundamental values)		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
TotalIssue	*Security Variable*	Total class issue. The total number of securities of this class quoted on the exchange. (Not calculated)
AssetBacking **NetTangibleAssets** **NTA**	*Security Variable*	Net Tangible Assets as last reported by the company, adjusted for dilution. (Not calculated)
EarningsPerShare **EPS**	*Security Variable*	Rolling 12 months earnings per share. (Not calculated)
DividendPerShare **DPS**	*Security Variable*	Rolling 12 months dividend rate per share. (Not calculated)
CurrentDividendAmount	*Security Variable*	Value of most recent dividend (net of withholding tax). (Not calculated)
FrankedPercent **Franked**	*Security Variable*	The percentage of the dividend on which tax has already been paid. (Not calculated)
GrossDividendAmount	*Security Variable*	Gross dividend amount. (Not calculated)
AnnualDividendPerShare	*Security Variable*	Total dividend rate per share over the last 12 months, including any special cash or scrip dividend. (Not calculated)
MarketCap	*Security Variable*	The current capitalisation of this security. (Calculated)
EarningsYield	*Security Variable*	Earnings per share as a percentage of price. (Calculated)
PERatio **PE**	*Security Variable*	Price Earnings Ratio (Calculated)
DividendYield	*Security Variable*	Dividends per share as a percentage of price. (Calculated)
DividendCover	*Security Variable*	Ratio between earnings and dividends. (Calculated)

Table 3 – Security Variables (aka fundamental values)		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
ExDate	*Security Variable*	Shares purchased after this date will not receive dividends. (Not calculated)
DividendPayableDate	*Security Variable*	Dividend payments are made on this date. (Not calculated)
IndustryCode GICS	*Security Variable*	8 digit GICS (Global Industry Classification System) industry code. (Not calculated)
Sector	*Security Variable*	Name of GICS sector. First of four levels of classification. (Not calculated)
IndustryGroup	*Security Variable*	Name of GICS industry group. Second of four levels of classification. (Not calculated)
Industry	*Security Variable*	Name of GICS industry. Third of four levels of classification. Not available for many securities. (Not calculated)
SubIndustry	*Security Variable*	Name of GICS sub-industry. Fourth of four levels of classification. Not available for most securities. (Not calculated)
SectorIndex	*Security Variable*	The S&P/ASX200 index symbol for the sector that the security belongs to. Useful when used in conjunction with LoadSymbol.

Table 4 – Attributes		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
Attributes are generally used to customise the appearance of a chart or plot. You can see this from the list below. However, this does not apply to some of the attributes including: Alias, Author, Citation, Description, etc. which tend to refer to the entire script and simply provide additional details, or a way to record various documentation items and refer to them later. Note the special notes below that apply to the categorisation of indicators and scans.		
Alias	Attribute	An alternative name by which the formula may be called within BullScript. This will often be an abbreviation.
Author	Attribute	The author of the indicator. By convention, use the last name then the first name. Multiple authors may be listed, separated by pipes. **NOTE**: For an indicator, this field enables the special lookup when inserting an indicator. For a scan, it automatically handles the grouping of scans in the BullScan Manager window.
Category	Attribute	The category that the formula is in. Multiple categories may be listed, separated by pipes. **NOTE**: For an indicator, this field enables the special lookup when inserting an indicator. For a scan, it automatically handles the grouping of scans in the BullScan Manager window.

Table 4 – Attributes		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
Citation	Attribute	A reference to documentation on which the formula is based.
Color	Attribute	The default colour of the plot.
Description	Attribute	A description of the formula. This will appear on the *Insert Indicator* dialogue box.
FillStyle	Attribute	Background pattern for *ribbons*.
Font	Attribute	The font face, or family, used for markers and text plots.
FontSize	Attribute	The size of the font in points for markers and text plots.
FontStyle	Attribute	The style of the font for markers and text plots.
HorzLine	Attribute	Levels at which horizontal lines should be displayed.
LineStyle	Attribute	The style with which the plot will be drawn.
LinkedChart	Attribute	If an indicator has this attribute and it is included in a scan, then when a linked chart is created, only selected plots are shown on the chart.
Name	Attribute	The name for an individual plot.
Password	Attribute	The password must be entered to view or edit the indicator script again - but not to use the indicator.
Period	Attribute	The timeframe that this indicator intended for use on. Do not specify if the indicator can be used on any timeframe.
SortPlots	Attribute	Normally an indicators plots are shown in the properties window in the order they are scripted. This attribute causes them to be sorted by name.
Target	Attribute	The pane and scale that the plot will use by default.
TextAlign	Attribute	Specifies where text is placed relative to a data point.
TextFormat	Attribute	Allows advanced formatting options for displaying dates and numbers as text.
Transparency	Attribute	Specifies the transparency for a filled linestyle. 0 for completely opaque and 100 for completely transparent.
Visible	Attribute	Whether or not the plot is visible by default.
Width	Attribute	The width of the plot line.

Table 5 – Functions — introduction		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
IF(test,expr1,expr2)	function	This function performs a test, and if true then performs one calculation **expr1** , and if false, performs an alternative calculation **expr2** .
Hist(expr,n)	Function	Calculates **expr** as at **n** bars ago. (Also see: **Hist**, **Future**, **Prev**)
Future( )	Function	Calculates **expr** as at **n** bars ahead from now. (Also see: **Hist**, **Future**, **Prev**)
Previous( ) **Prev( )**	Function	This often refers to the “previous” result, but if specifying the optional parameters, can be used for other purposes. (Also see: **Hist**, **Future**, **Prev**)

Table 6 – Functions — general		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
AllFalse(expr,n)	Function, General	This function calculates the expr for **every** bar over n bars. The result is either true or false. If the expr is evaluated to be *false* for every bar, then the result is **true**.
AllTrue(expr,n)	Function, General	This function calculates the expr for **every** bar over n bars. The result is either true or false.
AnyFalse(expr,n)	Function, General	This function calculates the expr for **every** bar over n bars, and looks for at least one *false* result. The result is either true or false.
AnyTrue(expr,n) **Alert( )**	Function, General	This function calculates the expr for **every** bar over n bars, and looks for at least one *true* result. The result is either true or false.
Barnumber	Function, General	This function calculates the total number of bars for the current security.
BarsSince(expr,n)	Function, General	This function calculates the expr (a boolean expression that will evaluate to true or false), and returns the number of bars since expr was true.
Expresssion( )	Function, General	The Expression function can be used to prompt for, and to receive, a value that can be used in a formula.
ExtFml( )	Function, General	*** **HERE** ***
False( )	Function, General	
FirstValue( )	Function, General	
Formula( )	Function, General	
Future( )	Function, General	(Also see: **Hist**, **Future**, **Prev**)

Table 6 – Functions — general		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`Hist( )`	Function, General	(Also see: **`Hist`**, **`Future`**, **`Prev`**)
`If( )`	Function, General	
`Input( )`	Function, General	
`InputMA( )`	Function, General	
`InputROC( )`	Function, General	
`InputSymbol( )`	Function, General	
`InputText( )`	Function, General	
`IsDefined( )`	Function, General	
`IsUndefined( )`	Function, General	
`LastValue( )`	Function, General	
`LoadSymbol( )`	Function, General	
`Param( )`	Function, General	
`Previous( )` **`Prev( )`**	Function, General	(Also see: **`Hist`**, **`Future`**, **`Prev`**)
`Requires( )`	Function, General	
`Trigonometric Functions`	Function, General	The standard trigonometric functions: Sin, Cos, Tan, etc.
`True`	Function, General	
`Undefined`	Function, General	
`ValueWhen( )`	Function, General	
`Wait( )`	Function, General	

Table 7 – Functions — mathematical		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`ABS( )`	Function, Mathematical	
`Add( )`	Function, Mathematical	
`Ceiling( )` **`Ceil( )`**	Function, Mathematical	
`Correl( )`	Function, Mathematical	
`Cum( )` **`Sub( )`**	Function, Mathematical	
`Div( )`	Function, Mathematical	
`Exp( )`	Function, Mathematical	
`Fibonacci( )`	Function, Mathematical	
`Floor( )`	Function, Mathematical	
`Frac( )`	Function, Mathematical	
`Int( )`	Function, Mathematical	

Table 7 – Functions — mathematical		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`LinearReg( )`	Function, Mathematical	
`LinRegSlope( )`	Function, Mathematical	
`Log( )`	Function, Mathematical	
`Max( )`	Function, Mathematical	
`MaxLinearReg( )`	Function, Mathematical	
`Median( )`	Function, Mathematical	
`Min( )`	Function, Mathematical	
`MinLinearReg( )`	Function, Mathematical	
`Mod( )`	Function, Mathematical	
`Mode( )`	Function, Mathematical	
`Mul( )`	Function, Mathematical	
`Neg( )`	Function, Mathematical	
`Pi`	Function, Mathematical	
`Power( )`	Function, Mathematical	
`Round( )` `Prec( )`	Function, Mathematical	
`Sign( )`	Function, Mathematical	
`Sqrt( )`	Function, Mathematical	
`StDev( )` `St( )`	Function, Mathematical	
`StEyx( )`	Function, Mathematical	
`Sub( )`	Function, Mathematical	
`Sub( )` `Cum( )`	Function, Mathematical	
`Trendline( )`	Function, Mathematical	
`Variance( )`	Function, Mathematical	

Table 8 – Functions — candle patterns		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`BearHarami`	***Function, Candle Pattern***	(lengthy list of candle patterns not included here yet)

Table 9 – Functions — pattern recognition		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`Cross( )`	Function, Pattern Recognition	
`DownTrend`		
`GapDown( )`	Function, Pattern Recognition	
`GapUp`	Function, Pattern Recognition	
`Inside`	Function, Pattern Recognition	
`Outside`	Function, Pattern Recognition	
`Peak( )`	Function, Pattern Recognition	
`PeakBars( )`	Function, Pattern Recognition	
`Trough( )`	Function, Pattern Recognition	
`TroughBars( )`	Function, Pattern Recognition	
`UpTrend`	Function, Pattern Recognition	

Table 10 – Functions — date and time		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`Date( )`	Function, Date and time	
`DateAdd( )`	Function, Date and time	
`DateDiff( )`	Function, Date and time	
`DayOfMonth( )`	Function, Date and time	
`DayOfWeek( )`	Function, Date and time	
`Hour( )`	Function, Date and time	
`InputDate( )`	Function, Date and time	
`Minute( )`	Function, Date and time	
`Month( )`	Function, Date and time	
`Now()`	Function, Date and time	
`OnOrSkipped( )`	Function, Date and time	

Table 10 – Functions — date and time		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`Second( )`	Function, Date and time	
`Year( )`	Function, Date and time	
`SystemDate( )`	Function, Date and time	

Table 11 – Functions — highest/lowest		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`Highest( )` `HHV( )`	Function, Highest/Lowest	
`Highest( )`	Function, Highest/Lowest	***(details not included yet)***
`HighestBars( )` `HHVBars( )`	Function, Highest/Lowest	***(details not included yet)***
`HighestSince( )`	Function, Highest/Lowest	***(details not included yet)***
`HighestSinceBars( )`	Function, Highest/Lowest	***(details not included yet)***
`Lowest( )` `LLV( )`	Function, Highest/Lowest	***(details not included yet)***
`LowestBars( )`	Function, Highest/Lowest	***(details not included yet)***
`LowestSince( )`	Function, Highest/Lowest	***(details not included yet)***
`LowestSinceBars( )`	Function, Highest/Lowest	***(details not included yet)***

Table 12 – Functions — indicator		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`ASwing( )`	Function, Indicator	**(Many indicator functions not included yet)**
`AD( )`		
`ADX( )`		
`ATR(N)`	Function, Indicator	**(Many indicator functions not included yet)**

Table 13 – Functions — text		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`Left( )`	Function, Text	***(details not included yet)***
`Right( )`	Function, Text	***(details not included yet)***
`SubString( )`	Function, Text	***(details not included yet)***
`ToLower( )`	Function, Text	***(details not included yet)***
`ToUpper( )`	Function, Text	***(details not included yet)***
`Trim( )`	Function, Text	***(details not included yet)***

Table 14 – Functions — TradeSim		
Function name, keyword, symbol, etc.	**Type**	**Brief description** *(See the BullScript Help for more details)*
`EntryTrigger`	Function, TradeSim	***(details not included yet)***
`EntryPrice`	Function, TradeSim	***(details not included yet)***
`ExtFml`	Function, TradeSim	The ExtFml allows BullScript to call functions in 3rd party plug-ins, such as the TradeSim DLL.