



# *iExpertAdvisor*<sup>TM</sup>



# ***FX TrendLine Plug-in***

Requires VTS-Connect minimum version **4.0.0.40**

The ***FX TrendLine Plug-in*** allows any Expert Advisor to detect if one or more manual or automatically drawn trend lines have been broken.

## ***What is a Plug-in?***

*VTS stands for Visual Traders Studio.*

*The VTS Expert Advisor Builder is a Windows graphical application that enables non-programmers to build complex Expert Advisors by dragging, dropping and connecting logical elements.*

*The VTS application contains basic functionality to build almost any Expert Advisor.*

*A VTS Plug-in allows traders to easily implement advanced trading techniques using an add-on user interface.*

## Contents

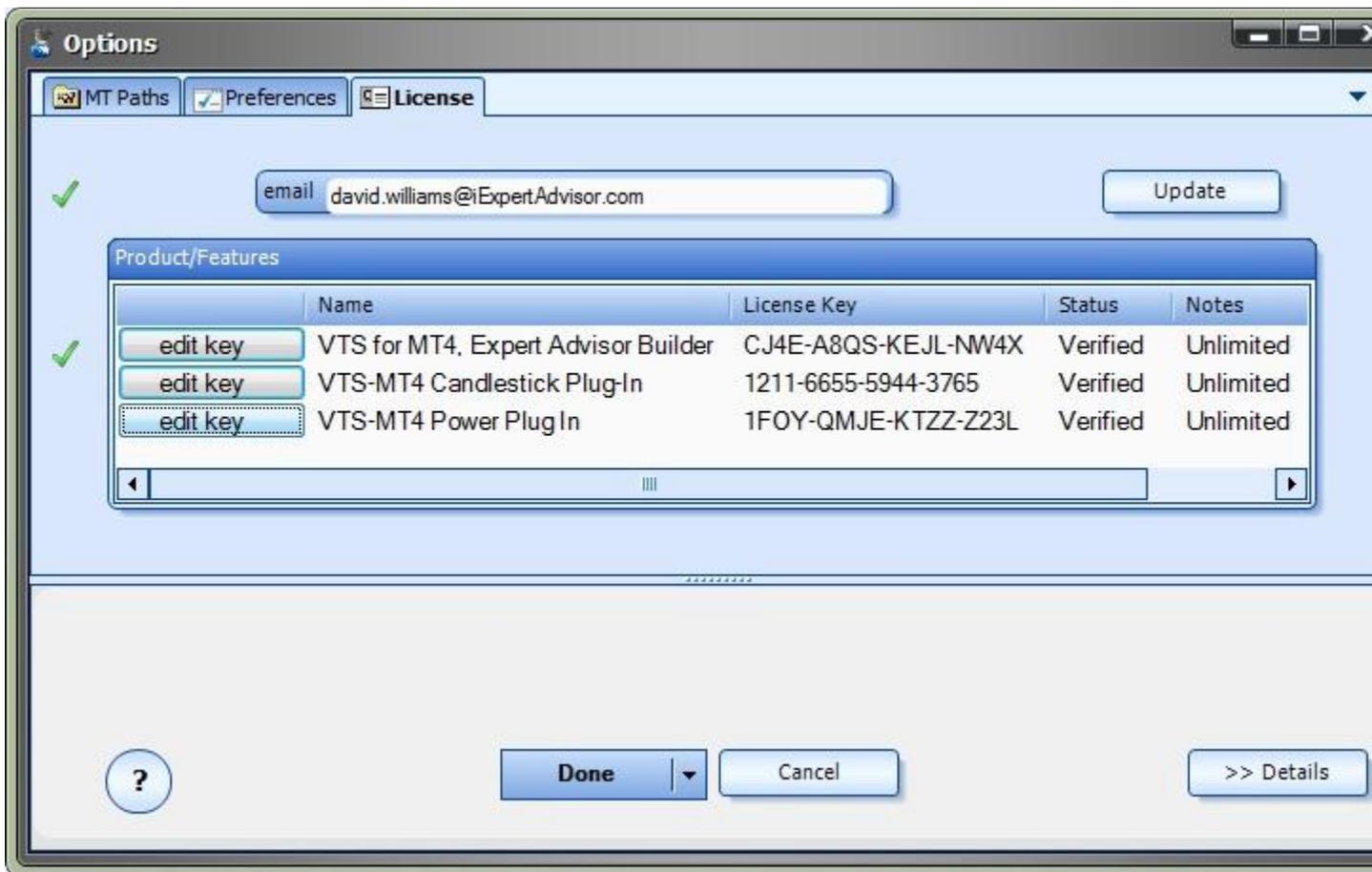
Enable the <i>FX TrendLine</i> Plug-in.....	4
TrendLines Functions in the Toolbox.....	5
TrendLine Functions.....	7
fnDrawTrendLine .....	8
fnIsTrendLineBroken.....	11
fnGetTrendLineValue.....	13
fnAddLabel .....	15
fnDeleteTrendLine .....	17
fnDeleteAllTrendLines.....	19
fnDeleteAllLabels .....	20
fnDeleteAllVerticalLines.....	21
Using FX TrendLine.....	22
Objects on the MetaTrader platform .....	22
Manually drawing a trend line .....	24
Removing a trend line (or any other Object) .....	25
Building an Expert Advisor to manage a manually drawn trend line .....	26
Building an Expert Advisor to draw and manage a trend line .....	29

## Enable the *FX TrendLine Plug-in*

You must enter your License key to enable the **FX TrendLine Plug-in**. Your license key for all of your VTS products can be found in the [Members Area](#).

License keys are entered in Visual Traders Studio (VTS) from the License entry tab.

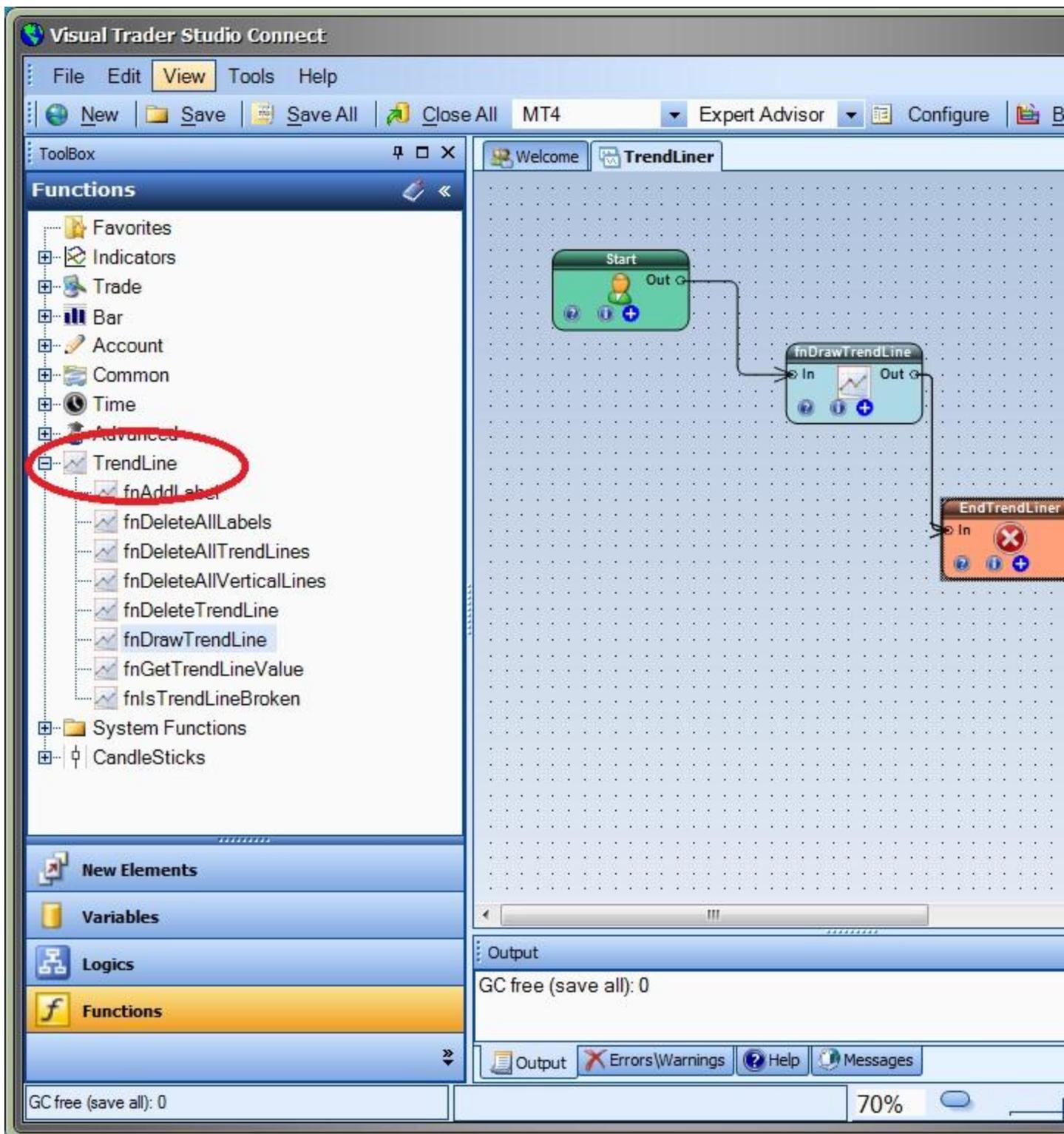
- The **email** address is the email address used to purchase [VTS](#).
- The **License Key** is the key that is sent to the email address.
- The **Verify** button is used to verify the email address and license key.
- The **Add** button is used to add a key.
- The **Remove** button is used to remove a key.
- Double-click a key to edit its value.



## **TrendLines Functions in the Toolbox**

Once enabled, the TrendLine functions are available in the [Toolbox](#) Function tab under the TrendLines menu.

These functions are dragged and dropped from the Toolbox onto the [Drawing Pad](#) like any other functions.



## TrendLine Functions

The TrendLine function library include these functions:

- [fnDrawTrendLine](#)
  - Used to programatically draw a trend line on a price chart.
- [fnIsTrendLineBroken](#)
  - Used to determine if a price value has broken through a trend line.
- [fnGetTrendLineValue](#)
  - Used to get the value of a trend line at any point on the line.
- [fnAddLabel](#)
  - Used to add a label to a price chart.
- [fnDeleteTrendLine](#)
  - Used to programatically delete a trend line from a price chart.
- [fnDeleteAllTrendLines](#)
  - Used to programatically delete all trend lines from a price chart.
- [fnDeleteAllLabels](#)
  - Used to programatically delete all labels from a price chart.
- [fnDeleteAllVerticalLines](#)
  - Used to programatically delete all vertical lines from a price chart.

## fnDrawTrendLine

The trend line function **fnDrawTrendLine** is used to programatically draw a trend line on a price chart.

After the **fnDrawTrendLine** function has been added to a Drawing, it is configured by clicking the (+) button along the bottom of the Element.

The [Function Configuration](#) window allows you to select values for each parameter.

Parameter Name	Data type	Description
Name	string	The name of the trend line. All trend lines are identified by name. The name should be unique. The pull-down menu provides sample names, but the name can be overwritten to any allowable name. The name must be surrounded by double quotes, for example, "myName". Avoid using blank spaces in the name as this may cause unpredictable behaviour.
ReDrawInterval	DRAW type (integer)	The interval at which the trend line is redrawn. The pull-down menu offers these choices: RD_ONCE : draw the line one time only RD_DAILY : redraw the line on each new day RD_HOURLY : redraw the line on each new hour RD_NEWBAR : redraw the line on each new bar RD_NEWTICK : redraw the line on each new tick
StartBar	integer	The horizontal location (bar) on the price chart where the trend line begins. Lines are drawn left to right, so the StartBar is to the left of the EndBar on a price chart. The StartBar is a larger number than the EndBar.
StartPrice	double	The vertical location (price) where the trend line begins. The StartPrice is the price value of the StartBar. The pull-down menu offers these choices: PRICE_CLOSE PRICE_OPEN PRICE_HIGH PRICE_LOW PRICE_MEDIAN PRICE_TYPICAL PRICE_WEIGHTED
StartOffset	integer	A positive or negative offset of the StartPrice, where the increment equals 1 point. This value moves the starting point of the line up or down vertically on the price chart.
EndBar	integer	The horizontal location (bar) on the price chart where the trend line ends. Lines are drawn left to right, so the EndBar is to the right of the StartBar on a price chart. The EndBar is a smaller number than the StartBar.

EndPrice	double	The vertical location (price) where the trend line ends. The EndPrice is the price value of the EndBar. The pull-down menu offers these choices: PRICE_CLOSE PRICE_OPEN PRICE_HIGH PRICE_LOW PRICE_MEDIAN PRICE_TYPICAL PRICE_WEIGHTED
EndOffset	integer	A positive or negative offset of the EndPrice, where an increment equals 1 point. This value moves the end point of the line up or down vertically on the price scale.
LineColor*	Color	The color of the trend line.
DrawLabel*	boolean	Draws a label on the chart that contains the trend line name.
DrawMarks*	boolean	Draws vertical lines on the chart on the Start and End bar locations.

\* These parameters are found on the [Advanced](#) tab.

... \fnDrawTrendLineMql.fe("TrendLineUpper",DRAW\_ONCE,25,PRICE\_CLOSE,0,1,PRI...

Configure Advanced Notes Message

elements\user\functions\system6\fnDrawTrendLineMql.fe Parameter 1 of 8

Name: TrendLineUpper

elements\user\functions\system6\fnDrawTrendLineMql.fe Parameter 2 of 8

ReDrawInterval: DRAW\_ONCE

elements\user\functions\system6\fnDrawTrendLineMql.fe Parameter 3 of 8

StartBar: 25

elements\user\functions\system6\fnDrawTrendLineMql.fe Parameter 4 of 8

StartPrice: PRICE\_CLOSE

elements\user\functions\system6\fnDrawTrendLineMql.fe Parameter 5 of 8

StartOffset: 0

elements\user\functions\system6\fnDrawTrendLineMql.fe Parameter 6 of 8

EndBar: 1

elements\user\functions\system6\fnDrawTrendLineMql.fe Parameter 7 of 8

EndPrice: PRICE\_CLOSE

elements\user\functions\system6\fnDrawTrendLineMql.fe Parameter 8 of 8

EndOffset: 0

Name:

fnDrawTrendLine

? Done OK >> Details

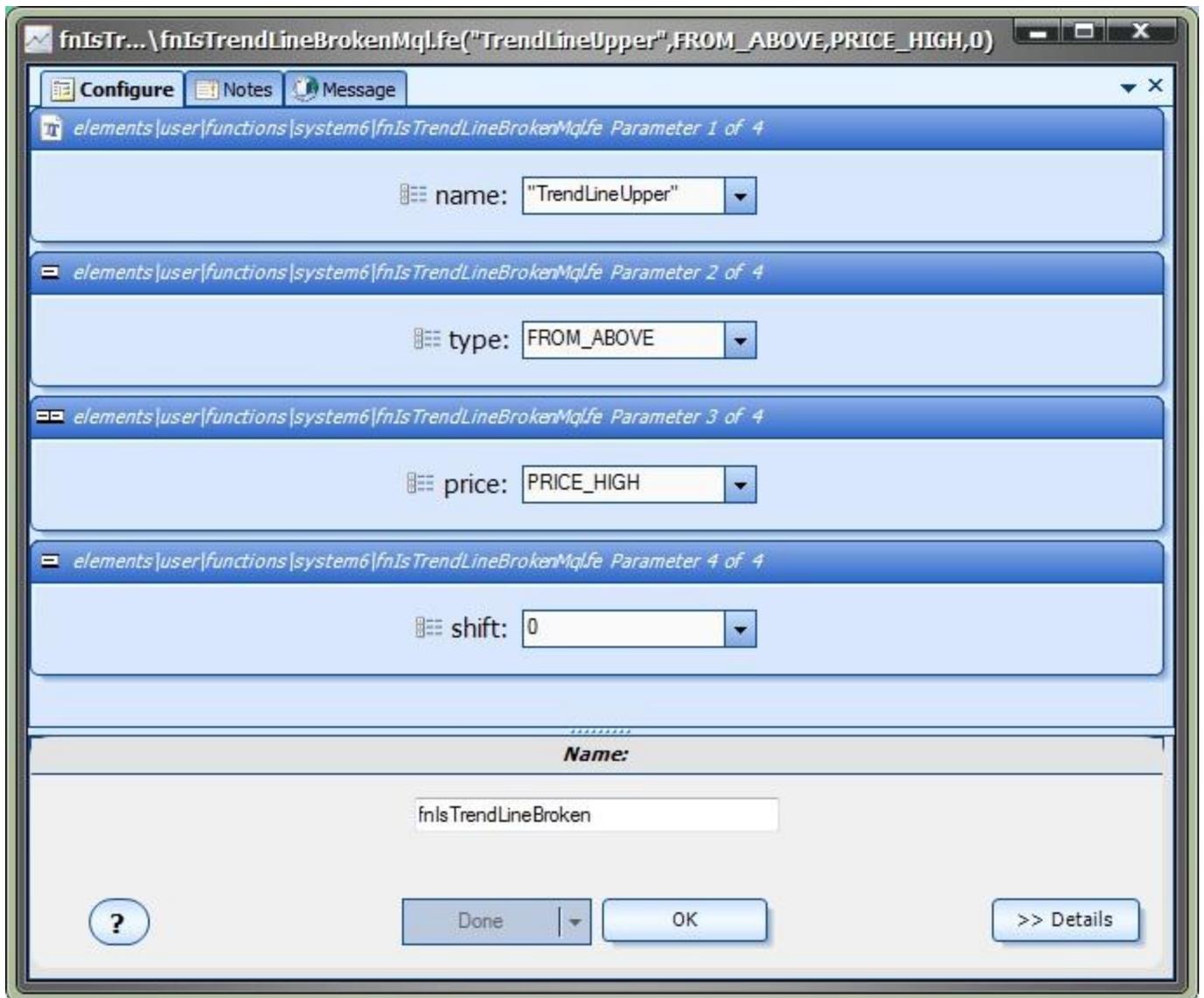
## fnIsTrendLineBroken

The **fnIsTrendLineBroken** function is used to determine if a price value has broken through a trend line.

After the **fnIsTrendLineBroken** function has been added to a Drawing, it is configured by clicking the (+) button along the bottom of the Element.

The [Function Configuration](#) window allows you to select values for each parameter.

Parameter Name	Data type	Description
name	string	The name of the trend line. All trend lines are identified by name. The name should be unique. The pull-down menu provides sample names, but the name can be overwritten to any allowable name. The name must be surrounded by double quotes, for example, "myName". Avoid using blank spaces in the name as this may cause unpredictable behaviour.
type	FROM value (integer)	The direction from which the price value was broken through the trend line. The pull-down menu offers these choices: FROM_ABOVE : The price value broke the trend line from above the line. FROM_BELOW : The price value broke the trend line from below the line.
price	double	The price value to use to determine if the price has broken through the trend line. The pull-down menu offers these choices: Ask Bid PRICE_CLOSE PRICE_OPEN PRICE_HIGH PRICE_LOW PRICE_MEDIAN PRICE_TYPICAL PRICE_WEIGHTED
shift	integer	The candle index on the price chart of where to start the trend line has been broken. Zero is the current forming candle, one is one candle to the left, etc.



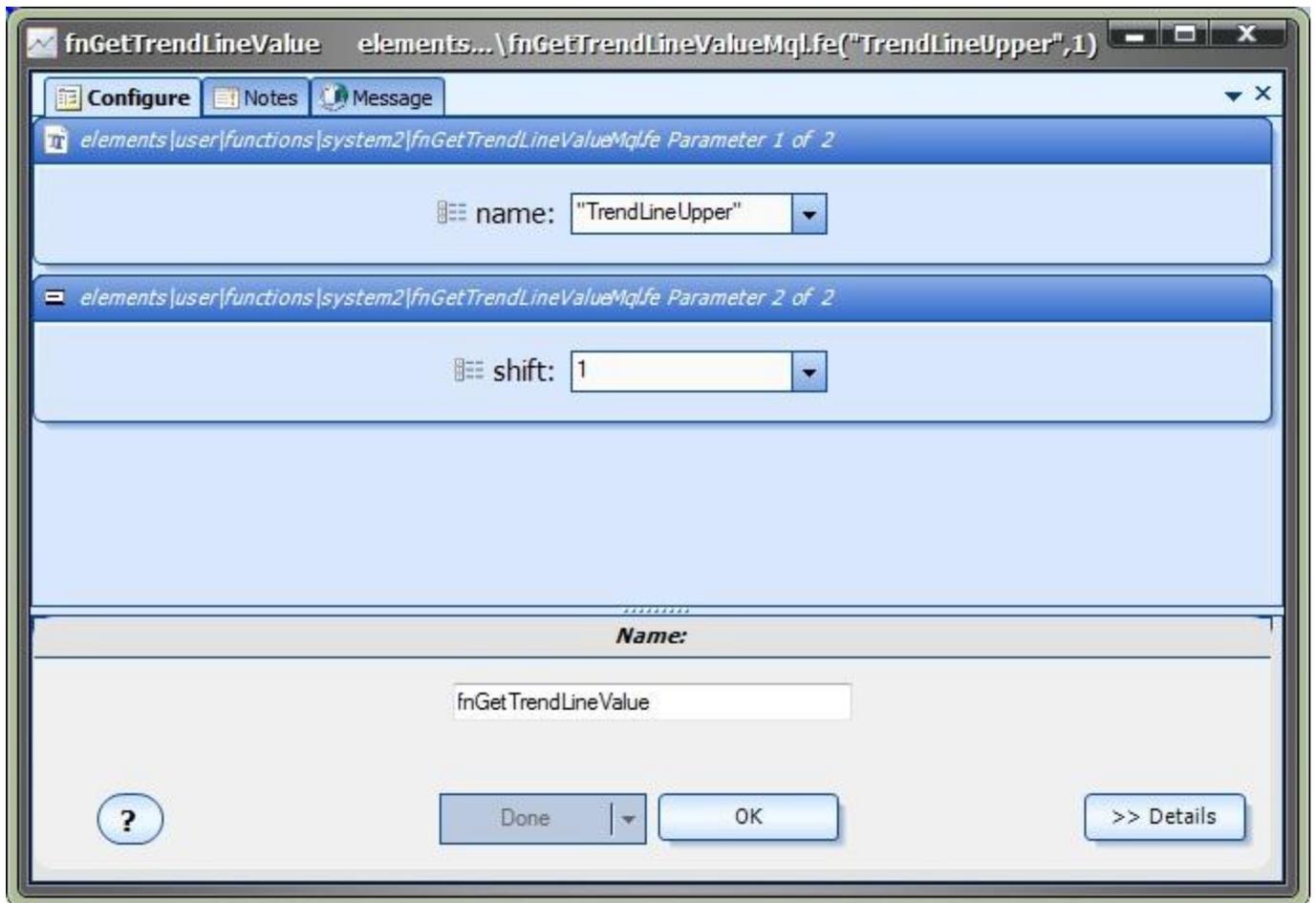
## fnGetTrendLineValue

The **fnGetTrendLineValue** function is used to get the value of a trend line at any point on the line.

After the **fnGetTrendLineValue** function has been added to a Drawing, it is configured by clicking the (+) button along the bottom of the Element.

The [Function Configuration](#) window allows you to select values for each parameter.

Parameter Name	Data type	Description
name	string	The name of the trend line. All trend lines are identified by name. The name should be unique. The pull-down menu provides sample names, but the name can be overwritten to any allowable name. The name must be surrounded by double quotes, for example, "myName". Avoid using blank spaces in the name as this may cause unpredictable behaviour.
shift	integer	The candle index on the price chart of where to calculate the trend line's value. Zero is the currently forming candle, one is one candle to the left, etc.



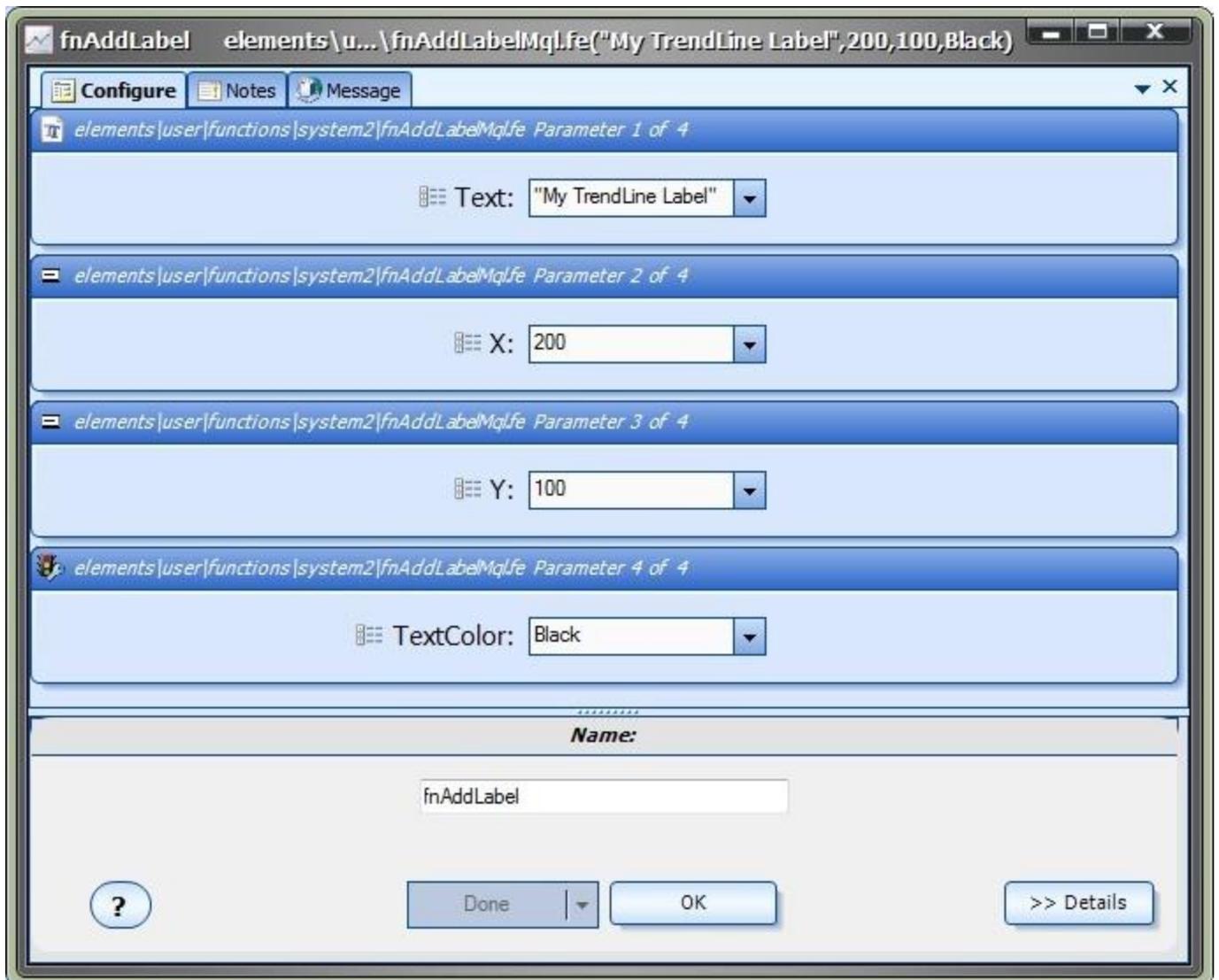
## fnAddLabel

The **fnAddLabel** is used to add a label to a price chart.

After the **fnGetTrendLineValue** function has been added to a Drawing, it is configured by clicking the (+) button along the bottom of the Element.

The [Function Configuration](#) window allows you to select values for each parameter.

Parameter Name	Data type	Description
Text	string	The text of the label to add to the price chart. Text must be surrounded by double quotes, for example "myName".
X	integer	The horizontal offset from the top left corner of the chart. The X value is zero at the far left corner of the chart and increases as you move to the right. The default value is 200.
Y	integer	The vertical offset from the top left corner of the chart. The Y value is zero at the top of the chart and increases as you move down. The default value is 100.
TextColor	Color	The color of the text.



## fnDeleteTrendLine

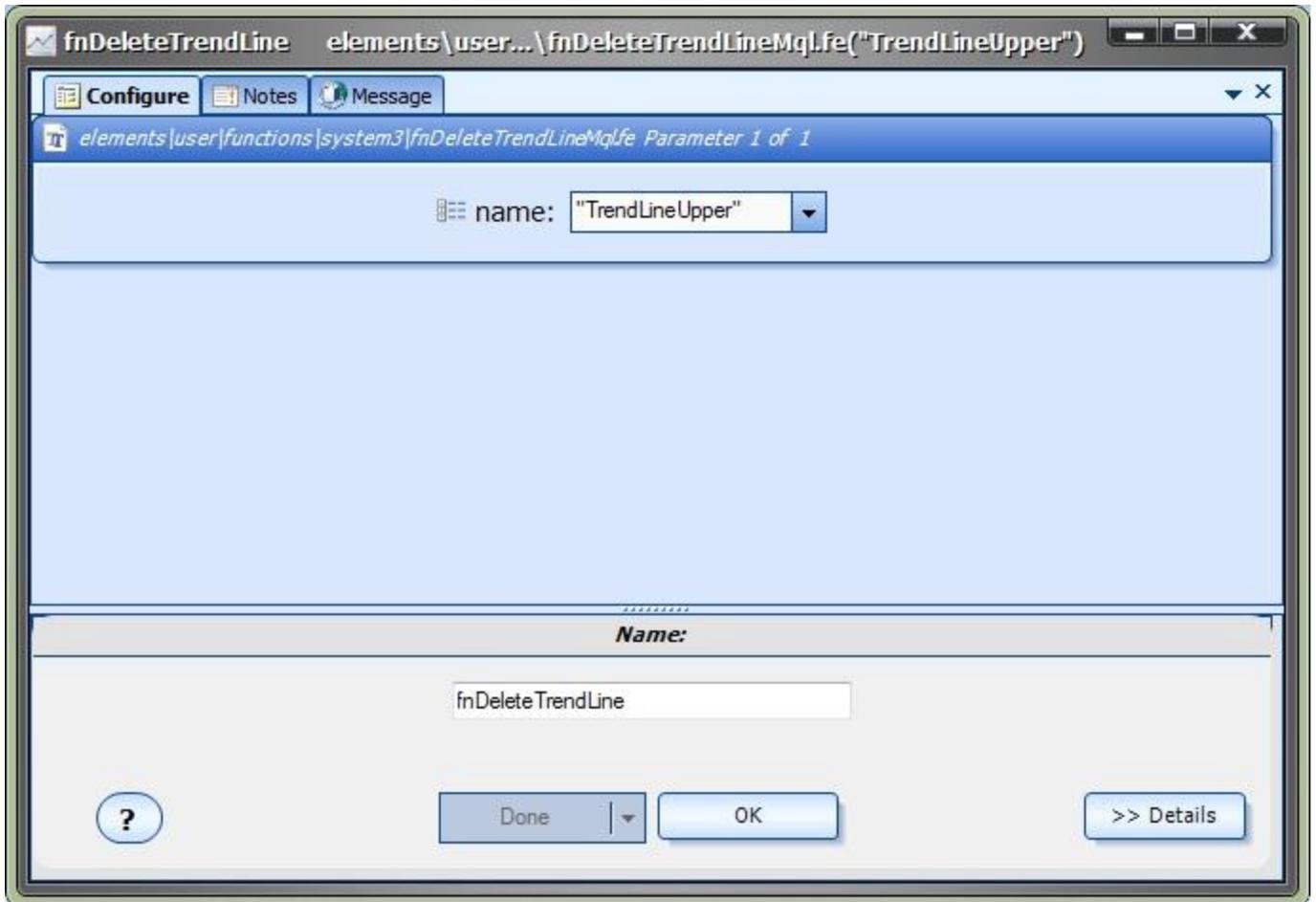
The **fnDeleteTrendLine** function is used to programmatically delete a trend line from a price chart.

After the **fnDeleteTrendLine** function has been added to a Drawing, it is configured by clicking the (+) button along the bottom of the Element.

The [Function Configuration](#) window allows you to select values for each parameter.

Parameter Name	Data type	Description
name	string	The name of the trend line. All trend lines are identified by name. The name should be unique. The pull-down menu provides sample names, but the name can be overwritten to any allowable name. The name must be surrounded by double quotes, for example, "myName". Avoid using blank spaces in the name as this may cause unpredictable behaviour.

Note: Any and all trend lines that match the name are deleted from the price chart. This includes trend lines created both manually and programmatically.



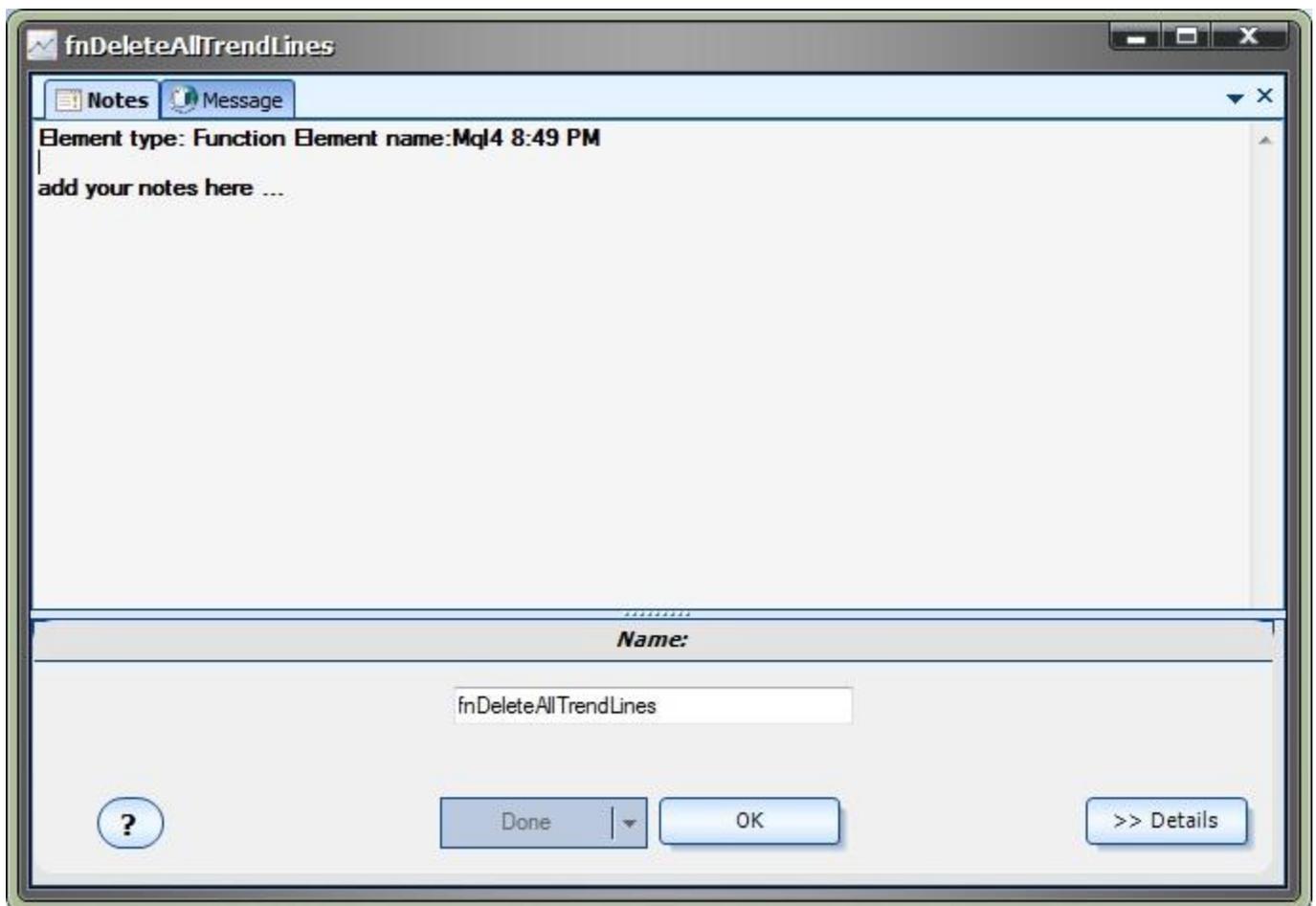
## fnDeleteAllTrendLines

The **fnDeleteAllTrendLines** function is used to programmatically delete all trend lines from a price chart.

After the **fnDeleteAllTrendLines** function has been added to a Drawing, it is configured by clicking the (+) button along the bottom of the Element.

The **fnDeleteAllTrendLines** function does not require any parameters.

Note: Any and all trend lines are deleted from the price chart. This includes trend lines created both manually and programmatically.



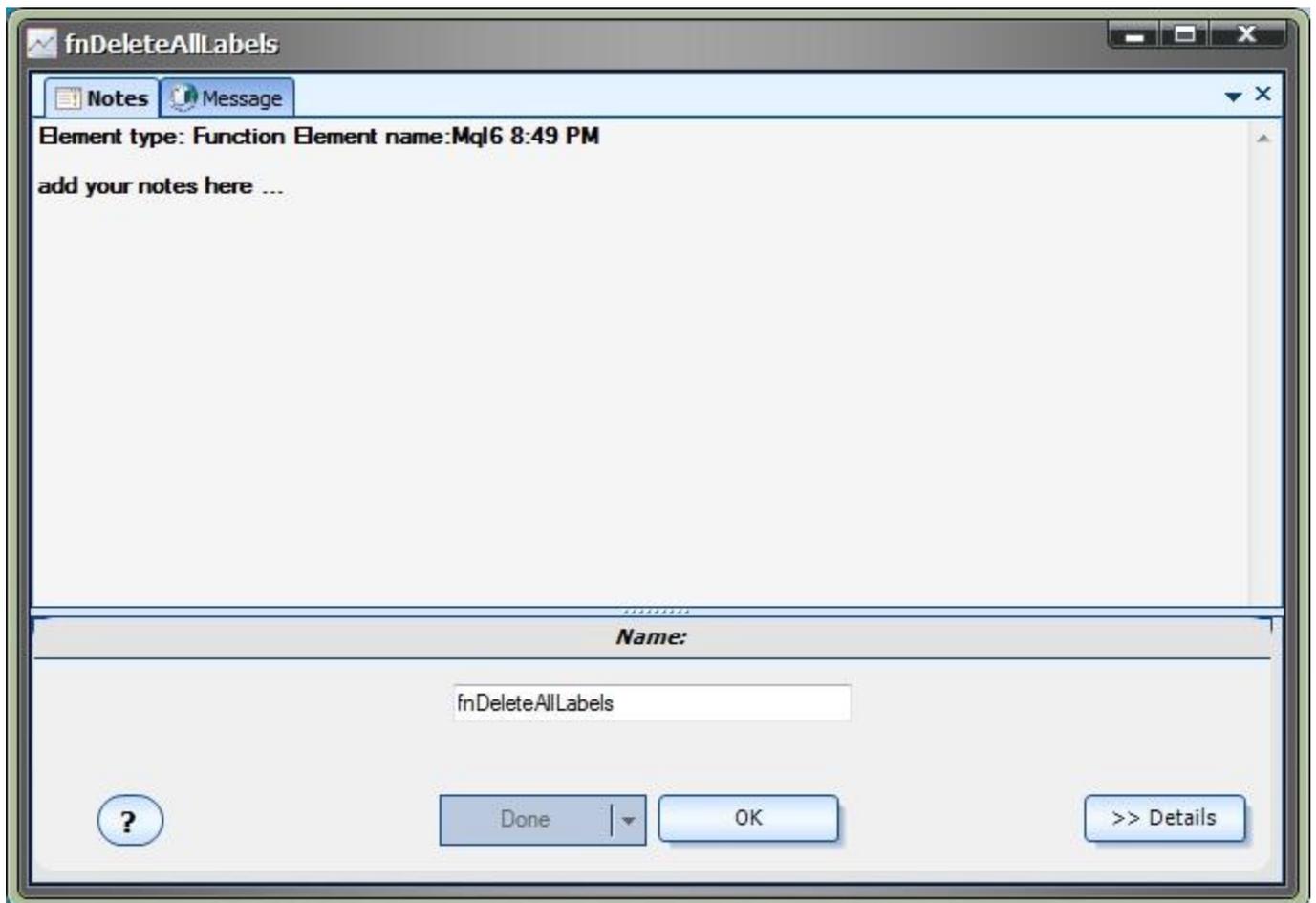
## fnDeleteAllLabels

The **fnDeleteAllLabels** is used to programatically delete all labels from a price chart.

After the **fnDeleteAllLabels** function has been added to a Drawing, it is configured by clicking the (+) button along the bottom of the Element.

The **fnDeleteAllLabels** function does not require any parameters.

Note: Any and all labels are deleted from the price chart. This includes labels created both manually and programatically.



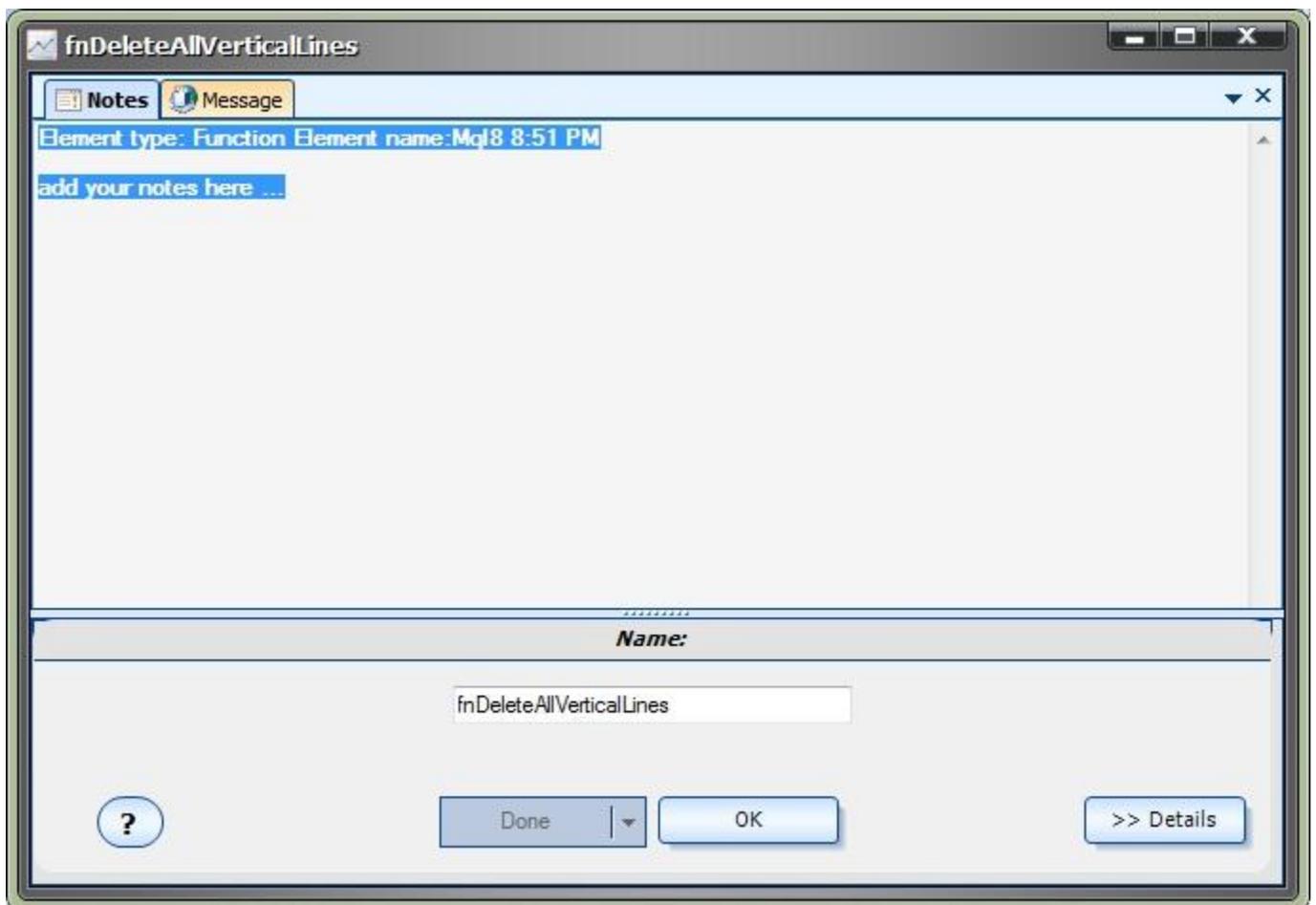
## fnDeleteAllVerticalLines

The **fnDeleteAllVerticalLines** function is used to programmatically delete all vertical lines from a price chart.

After the **fnDeleteAllVerticalLines** function has been added to a Drawing, it is configured by clicking the (+) button along the bottom of the Element.

The **fnDeleteAllVerticalLines** function does not require any parameters.

Note: Any and all vertical lines are deleted from the price chart. This includes vertical lines created both manually and programmatically.



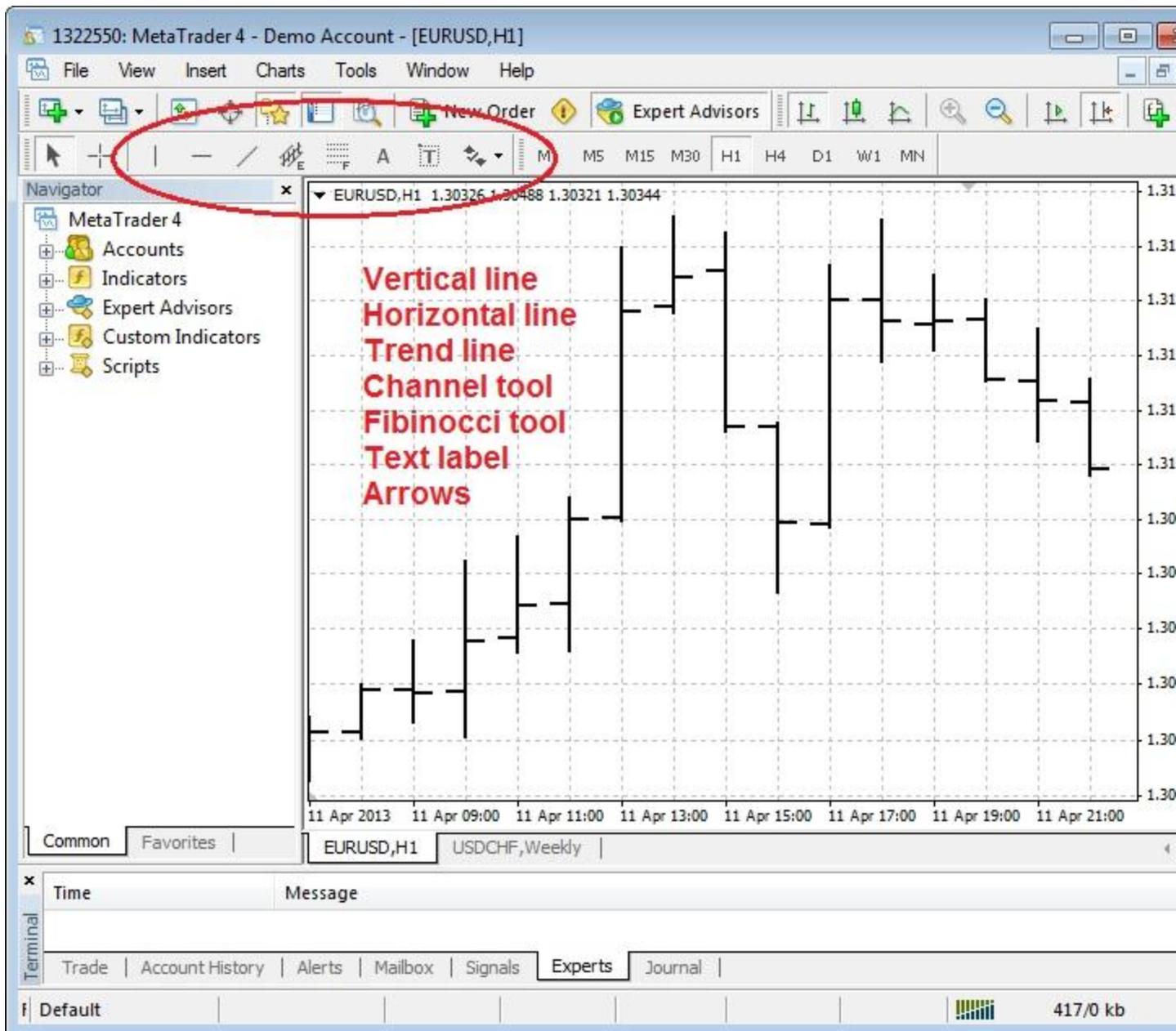
## Using FX TrendLine

### Objects on the MetaTrader platform

Every line, label and arrow that you see on a MetaTrader price chart is referred to as an Object. The MetaTrader platform provides a set of MQL functions for programatically creating and deleting Objects. The Object functions are available from the VTS [Function Toolbox](#) under the Advanced->Object menu. The Object functions are on the Advanced menu because they require advanced MQL knowledge and can be difficult to use.

Objects can be manually created using the toolbar on the top of the MetaTrader platform. The objects that can be created from the MT tool bar are:

- Vertical line
- Horizontal line
- Trend line
- Channels line
- Fibonacci lines
- Text
- Arrows



## Manually drawing a trend line

To manually draw a trend line on a MetaTrader price chart, click the trend line button, then click the start location on the chart, hold the mouse down while moving to the end location and release the mouse.

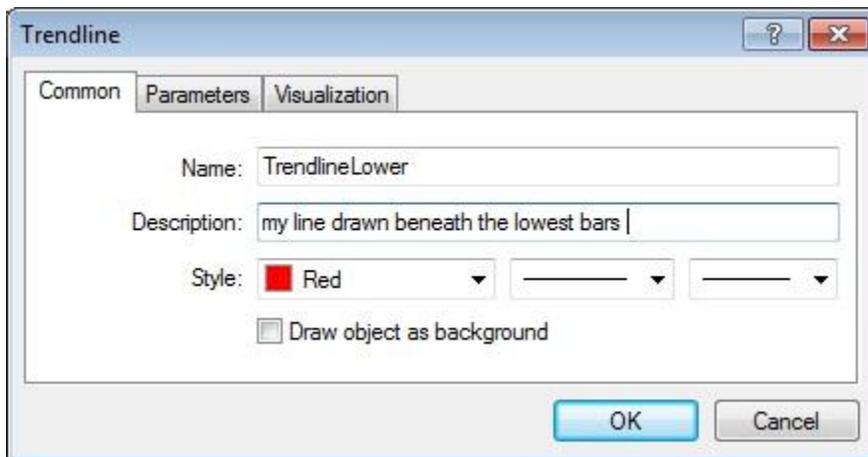
Once drawn, the location and angle of the trend line can be adjusted by double-clicking the line to select it and then moving it with the mouse.

When a trend line is manually created, it is given a generated name by the MetaTrader platform. For example, it may be named "Trendline 29693".

The **FX TrendLine Plug-in** identifies trend lines by their name. To change the name of a manually drawn trend line, double-click the trend line to select it, right-click the mouse and select "Trendline properties ...".

This will allow changing the trend line properties including the name.

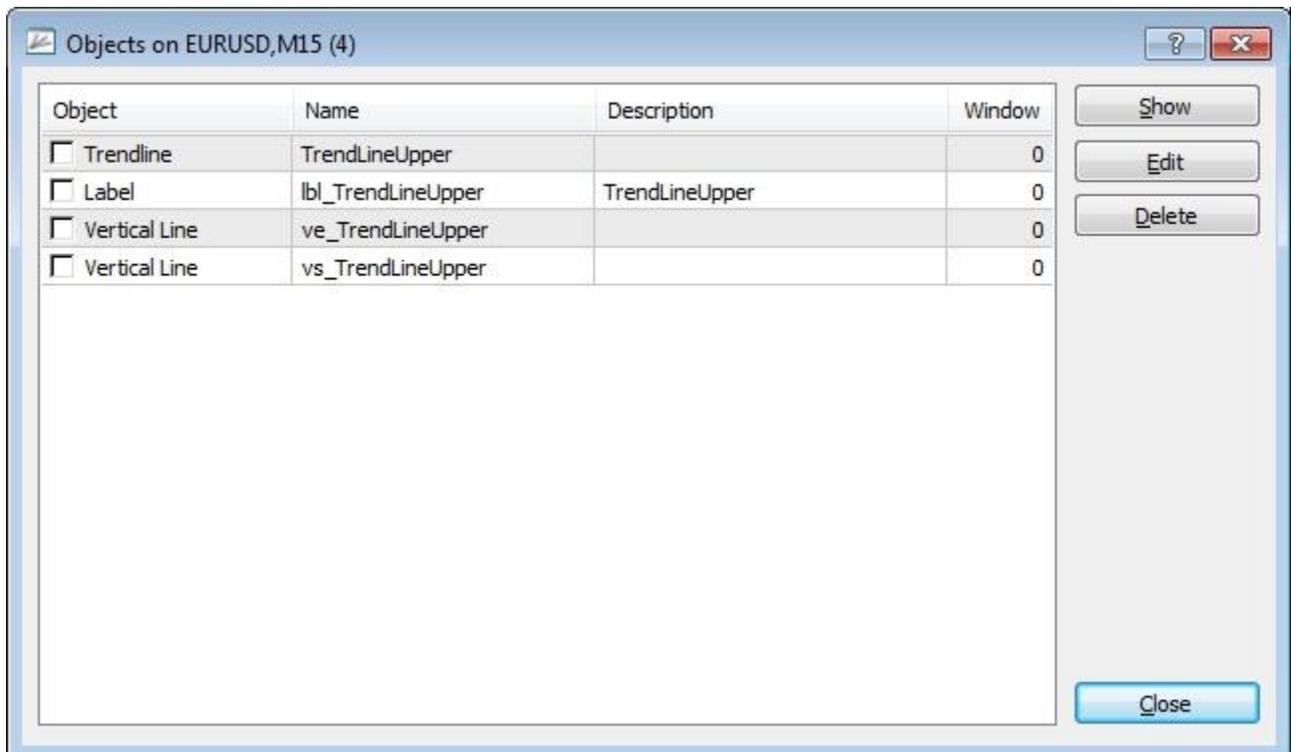
Note: The name of each trend line should be unique. Do not place more than one trend line on the same chart with the same name. However, it's permissible to have the same name trend line on different charts.



## Removing a trend line (or any other Object)

To manually remove a single trend line from a MetaTrader price chart, double-click the trend line to select it, and press the delete key on the keyboard.

Objects on a chart can be managed by going to **Charts->Objects->Object List**. This is the best method to delete many objects at once.



## Building an Expert Advisor to manage a manually drawn trend line

The **FX TrendLine Plug-in** can monitor any trend line on a chart. It does this by searching for a trend line by name.

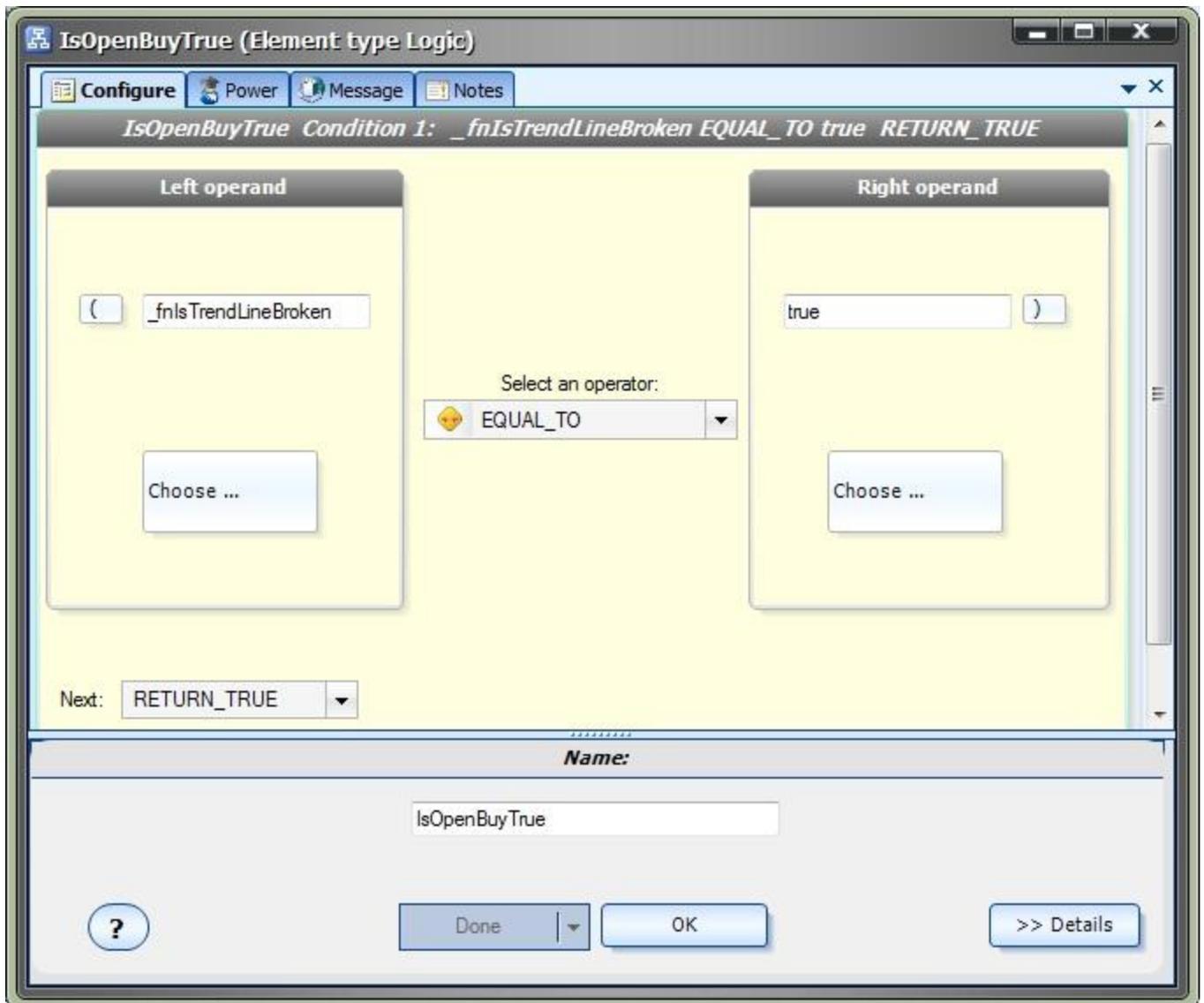
After you manually [draw your trend line](#) on your price chart, record the exact name of the trend line. This name will be referenced in your Expert Advisor by the [\*\*\*fnIsTrendLineBroken\*\*\*](#).

To build an Expert Advisor that opens a BUY trade when a trend line has been broken, drag and drop the TrendLine function ***fnIsTrendLineBroken*** on to the ***OpenBuyOrder*** drawing pad and connect it before the [Logic](#) element ***IsOpenBuyTrue***.

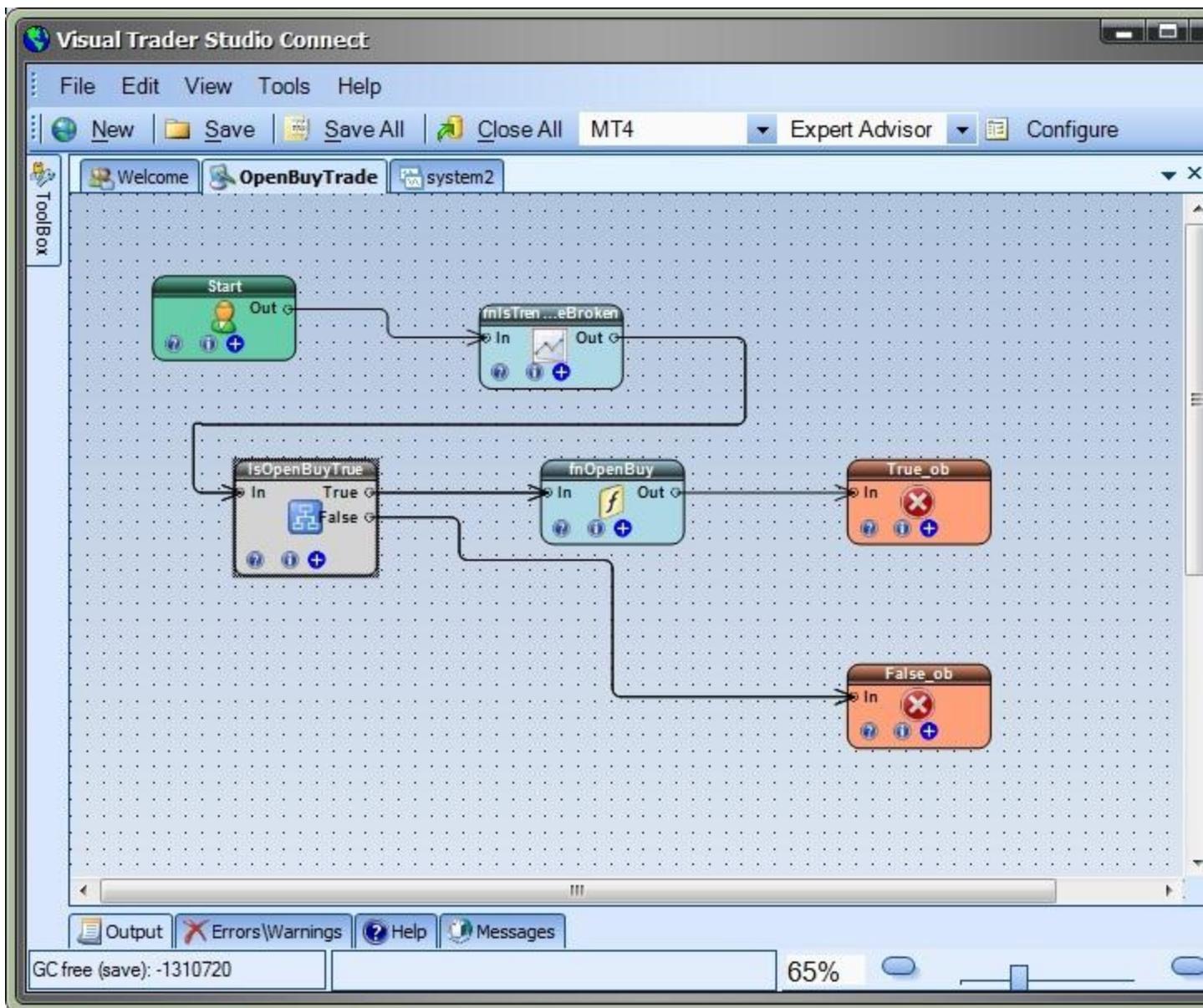
Set the [parameters](#) of the ***fnIsTrendLineBroken*** function:

- ***name***: set the value of the ***name*** parameter to the exact name of the trend line. For example "mytrendline".
- ***type***: set the value of the ***type*** parameter to the direction to test for the trend line break, either FROM\_ABOVE or FROM\_BELOW.
- ***price***: set the value of the ***price*** parameter to one of the [price constants](#), or ***Bid*** or ***Ask***.
- ***shift***: set the value of the ***shift*** parameter to the index value of the candle (or bar) to test for the break.
  - Normally, the ***shift*** value is set to 0 and the price value is set to ***Bid*** or ***Ask***. This will test the latest price against the trend line on the far right edge of the chart.
  - Alternatively, any candle on the chart can be tested for a break.

After connecting and configuring the ***fnIsTrendLineBroken*** function, configure the ***IsOpenBuyTrue*** Logic element to test for the break. The function ***IsOpenBuyTrue*** returns a value of ***true*** if/when the line is broken.



The full **OpenBuyOrder** drawing is shown below.



## Building an Expert Advisor to draw and manage a trend line

The **FX TrendLine Plug-in** can programatically draw a trend line on a price chart using the **fnDrawTrendLine** function.

To build an Expert Advisor that opens a BUY trade when a programatically drawn trend line has been broken, drag and drop the TrendLine function **fnDrawTrendLine** on to the **OpenBuyOrder** drawing pad and connect it before the [Logic](#) element **IsOpenBuyTrue**.

Set the [parameters](#) of the **fnDrawTrendLine** function:

- **Name:** set the value of the **name** parameter to the name of the trend line. For example "mytrendline". This name will be referenced later in the **fnIsTrendLineBroken** function.
- **ReDrawInterval:** set the value of the **ReDrawInterval** parameter to desired redraw value:
  - RD\_ONCE : draw the line one time only
  - RD\_DAILY : redraw the line on each new day
  - RD\_HOURLY : redraw the line on each new hour
  - RD\_NEWBAR : redraw the line on each new bar
  - RD\_NEWTICK : redraw the line on each new tick

Note: A trend line is projected to the right *indefinitely*. It is not necessary to redraw the trend line unless the drawing coordinates change.

Four coordinates are required to draw a line in two dimensional space. For example, on a typical graph, using the [Cartesian](#) coordinate system, the coordinates are usually defined as (x1, y1) and (x2,y2).

The primary coordinates of the trend line are defined as the **StartBar**, the **StartPrice**, the **EndBar** and the **EndPrice**.

- **StartBar:** set the value of the **StartBar** parameter to the bar (or candle) where the trend line should begin.
  - **NOTE:** *On a MetaTrader price chart, the currently forming candle is defined as candle 0 and the candle numbers increase to the left. Therefore, the StartBar is a larger number than the EndBar!*
- **StartPrice:** set the value of the **StartPrice** parameter to a [price value constant](#). The price is the vertical location where the line begins. The **StartPrice** is the *price value of the StartBar*.
- **StartOffset:** set the value of the **StartOffset** parameter to a positive or negative integer value (for example:10, 20, ...). The **StartOffset** value moves the vertical start location up or down relative to the **StartPrice**.
- **EndBar:** set the value of the **EndBar** parameter to the bar (or candle) where the trend line should end.
  - NOTE: A trend line is projected to the right *indefinitely* regardless of where the

EndBar is located. The EndBar simply defines the angle of the line, not its actual end point.

- **EndPrice:** set the value of the **EndPrice** parameter to a [price value constant](#). The price is the vertical location where the trend line ends. The **EndPrice** is the *price value of the EndPrice*.
  - NOTE: If the **EndPrice** is set as the Close, High, or Low of bar number 0, or to Ask or Bid, then the **EndPrice** may change value on each tick.
- **EndOffset:** set the value of the **EndOffset** parameter to a positive or negative integer value (for example:10, 20, ...). The **EndOffset** value moves the vertical end location up or down relative to the **EndOffset**.
- **LineColor:** optionally set the value of the **LineColor** parameter to a Color. The default is Black.
- **DrawLabel:** optionally set the value of the **DrawLabel** parameter to a true. (The default is false.) This will draw a label with the name of the trend line on the price chart. The label can be removed programatically using [fnDeleteAllLabels](#), or manually from from the [MetaTrader tool bar](#).
- **DrawMarks:** optionally set the value of the **DrawMarks** parameter to a true. (The default is false.) This will draw a vertical lines through the start and end bars. The vertical lines can be removed programatically using [fnDeleteAllVerticalLines](#), or manually from from the [MetaTrader tool bar](#).

