Strategy Optimizer Guide

Table Of Contents

Introduction

Disclaimer

The information and tools provided by PineScriptStrategy.com are for educational and informational purposes only. They should not be construed as financial advice, and are not intended to be a substitute for professional financial advice.

PineScriptStrategy.com does not guarantee the accuracy, completeness, timeliness, reliability, suitability, or availability of the information and tools provided. PineScriptStrategy.com will not be liable for any errors, omissions, or delays in this information or for any actions taken in reliance on this information.

PineScriptStrategy.com does not endorse or recommend any specific investments, strategies, or financial products. Any decisions you make based on the information and tools provided by PineScriptStrategy.com are your sole responsibility and at your own risk. You should carefully consider your own financial situation and consult with a financial advisor before making any financial decisions.

PineScriptStrategy.com is not responsible for any losses, damages, or other liabilities that may arise from the use of the information and tools provided. You agree to indemnify and hold PineScriptStrategy.com and its affiliates, employees, agents, and contractors harmless from any claims, damages, or expenses that may arise in connection with your use of the information and tools provided.

By using the information and tools provided by PineScriptStrategy.com, you acknowledge and agree to the terms of this financial disclaimer.

About The Author

Hi, I'm Paul.

I'm a trader and programmer with a passion for finance. Since 2015, I've been using Pine Script to create efficient and effective trading strategies. I've gained recognition within the Pine Script community through features on TradingView, Pine Coders, Udemy, and Upwork as a top performer in the field. In addition to my own trading pursuits, I also enjoy sharing my knowledge and expertise with others through my website, pinescriptstrategy.com. My ultimate goal is to empower individuals to take control of their wealth and break free from economic slavery, paving the way for a better world for all.

Download The Strategy Optimizer

Get it here --> Strategy Optimizer Code

Info

If the link no longer works, it means I have updated the tool and this document is outdated.
You will have to resubmit your information at pinescriptstrategy.com

Thank you for downloading the Strategy Optimizer! In this guide, we will go through the steps of setting up and using the tool on TradingView.

What Is This Tool?

As the name suggests, this tool is designed to streamline and optimize the strategy testing process in Pine Script. Whether you are a seasoned pro or new to the platform, this tool will save you time, hassle, and resources.

This workbook offers a step-by-step guide on using the Strategy Optimizer to backtest indicators without the need to manually convert your indicator to a strategy and write complex rules. Let the Strategy Optimizer handle the heavy lifting, so you can focus on developing the best trading strategies possible.

What is Pine Script?

Pine Script is a programming language developed by TradingView for use in creating custom technical indicators and trading strategies for financial charts on the TradingView platform. It is based on the Easy Language programming language, which was originally developed for use with the TradeStation platform. Pine Script is a relatively simple and easy-to-learn language, making it accessible to traders and investors who want to create custom studies and strategies for their own use or for sharing with others.

One of the limitations of Pine Script is that it can only be used on the TradingView platform. This means that scripts written in Pine Script cannot be used on other charting or trading platforms. Additionally, Pine Script has certain limitations in terms of the types of studies and strategies that can be created. For example, it is not possible to create custom studies or strategies that use real-time data or that execute trades automatically. These limitations may be frustrating for some users who want to create more advanced or sophisticated studies and strategies.

What is Tradingview?

TradingView is a financial platform that offers real-time stock and cryptocurrency market data, as well as technical analysis tools and user-generated content for individual investors. It is a popular destination for traders and investors to share their charts and ideas, and to learn from each other. The platform also offers a range of paid subscription services, such as access to advanced charting tools and real-time market data feeds.

Whats the difference between an indicator and a strategy?

In the context of TradingView and technical analysis, an indicator and a strategy are two different types of tools that are used to analyze and make decisions about the market. Here are the key differences between indicators and strategies:

- **Purpose:** Indicators are tools that are used to analyze the market and identify trends, patterns, and other characteristics of the data. They typically use mathematical formulas and algorithms to process the data and generate signals or outputs that can be used by traders to make decisions. Strategies, on the other hand, are tools that are used to automate the process of trading. They typically use a set of rules and logic to determine when to enter and exit trades, as well as how to manage the trades once they are open.

- **Functionality:** Indicators and strategies have different functionalities and capabilities. Indicators are typically designed to provide a specific type of analysis or signal, such as identifying trend direction, overbought/oversold conditions, or support/resistance levels. Strategies, on the other hand, are designed to execute trades and manage positions based on a set of rules and logic. This means that strategies can include indicators as part of their logic, but indicators cannot execute trades or manage positions on their own.

- **Implementation:** Indicators and strategies are implemented differently in Pine Script. Indicators are typically implemented as functions that take the data series as input and return the output (e.g. the signal) as output. Strategies, on the other hand, are implemented as collections of rules and logic that are executed by the Pine Script runtime environment. This means that strategies can include multiple indicators and other components, whereas indicators are typically standalone functions.

  Tldr

  In summary, indicators and strategies are both tools used for trading and technical analysis, but they serve distinct purposes. Indicators are used to analyze market data and generate signals, while strategies automate the process of trading. Indicators do not inherently provide backtesting results, making them more resource-efficient and faster to run. As a result, it can be beneficial to use indicators whenever possible.

Why Should I use this tool?

The Strategy Optimizer is a valuable tool for any trader looking to streamline their strategy testing process and make more informed decisions. It saves time and effort by separating indicator logic from strategy logic, resulting in code that is easier to manage and a single codebase for strategies. This allows you to experiment with multiple indicators and strategies without getting overwhelmed by excess code.

In addition to saving time and effort, the Strategy Optimizer can also help you make more informed trades. By backtesting your signals using a strategy, you can gain a deeper understanding of your indicators and set more accurate expectations for how they will perform in live trading scenarios. This can help you make more confident and informed trades, increasing your chances of success.

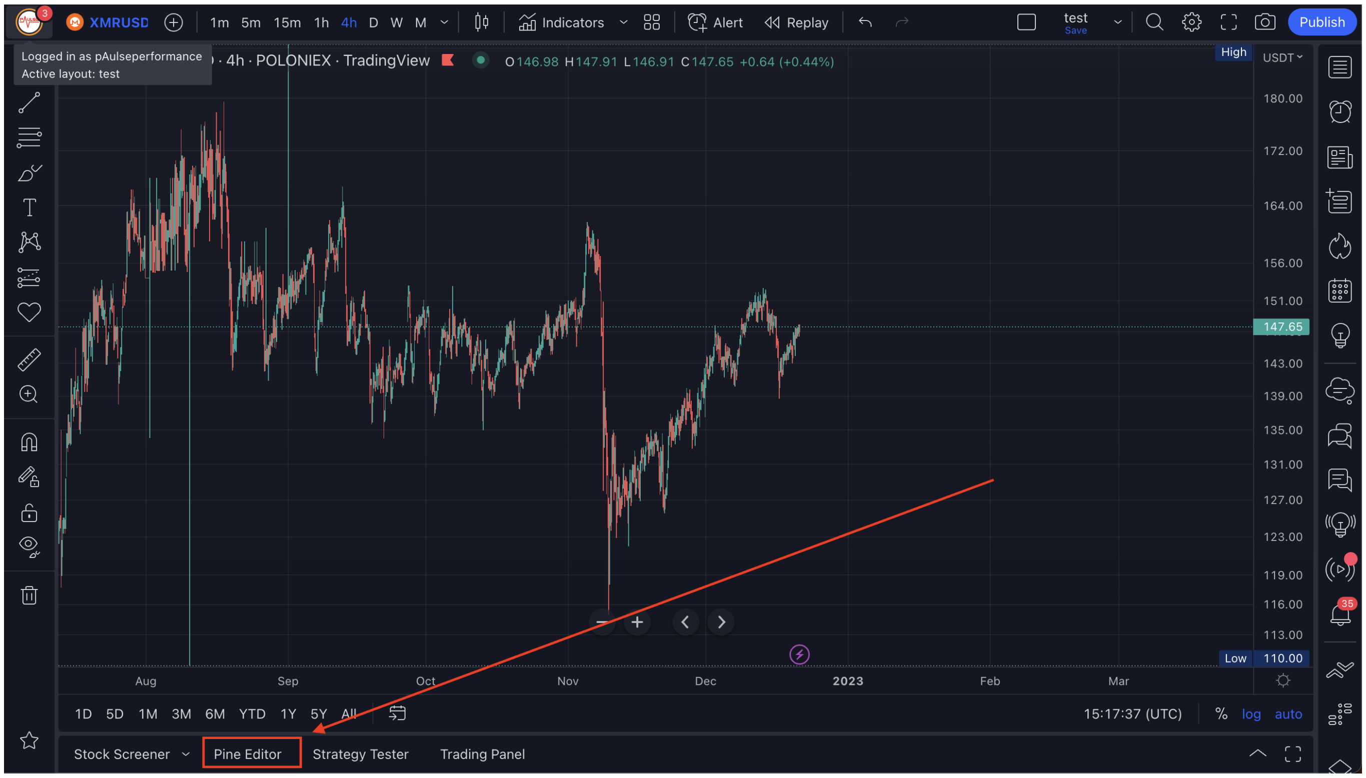
Overall, the Strategy Optimizer is a valuable tool for any trader looking to streamline their strategy testing process and make more informed decisions. It saves time and effort, improves understanding and interpretation of indicator signals, and facilitates experimentation with multiple indicators and strategies without excess code. Give it a try and see the difference it can make for your trading.

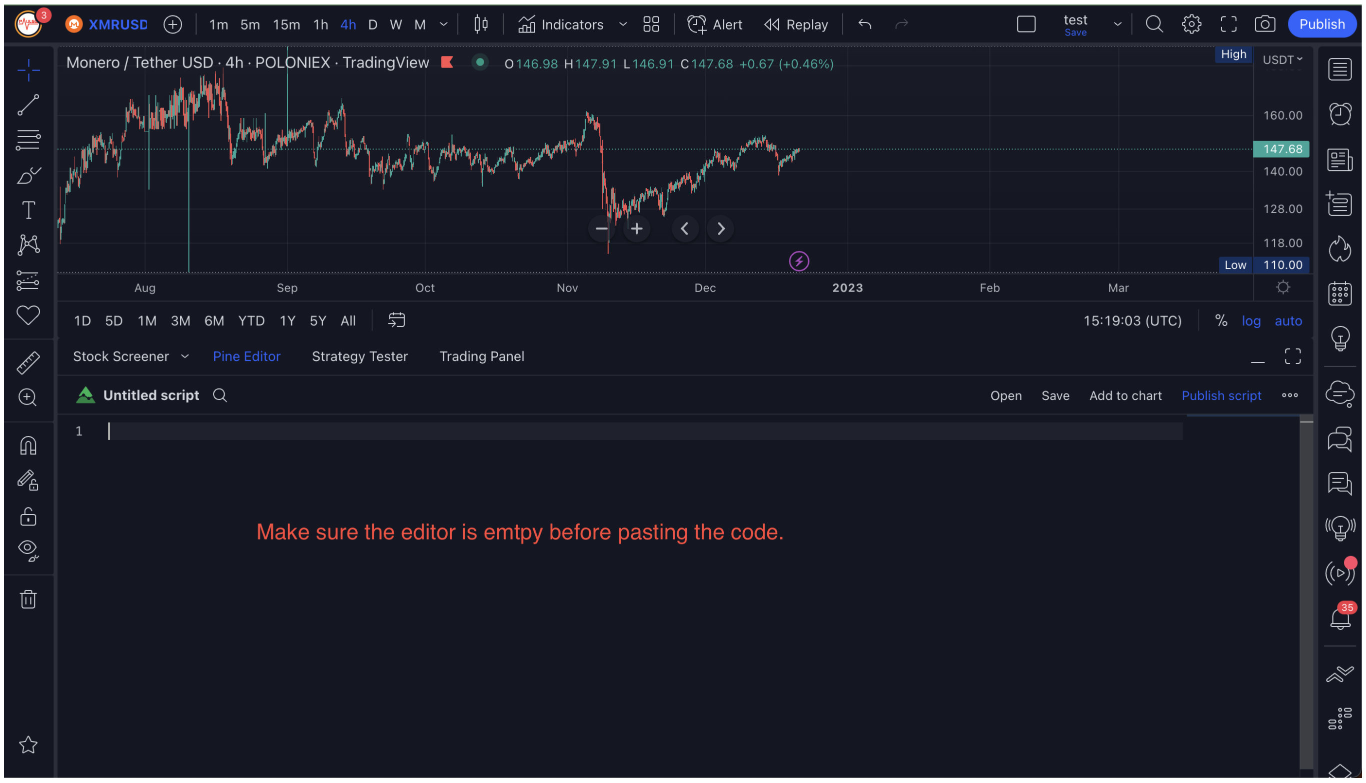
How Do I Use This Tool?

To get started, you will need to follow these steps:

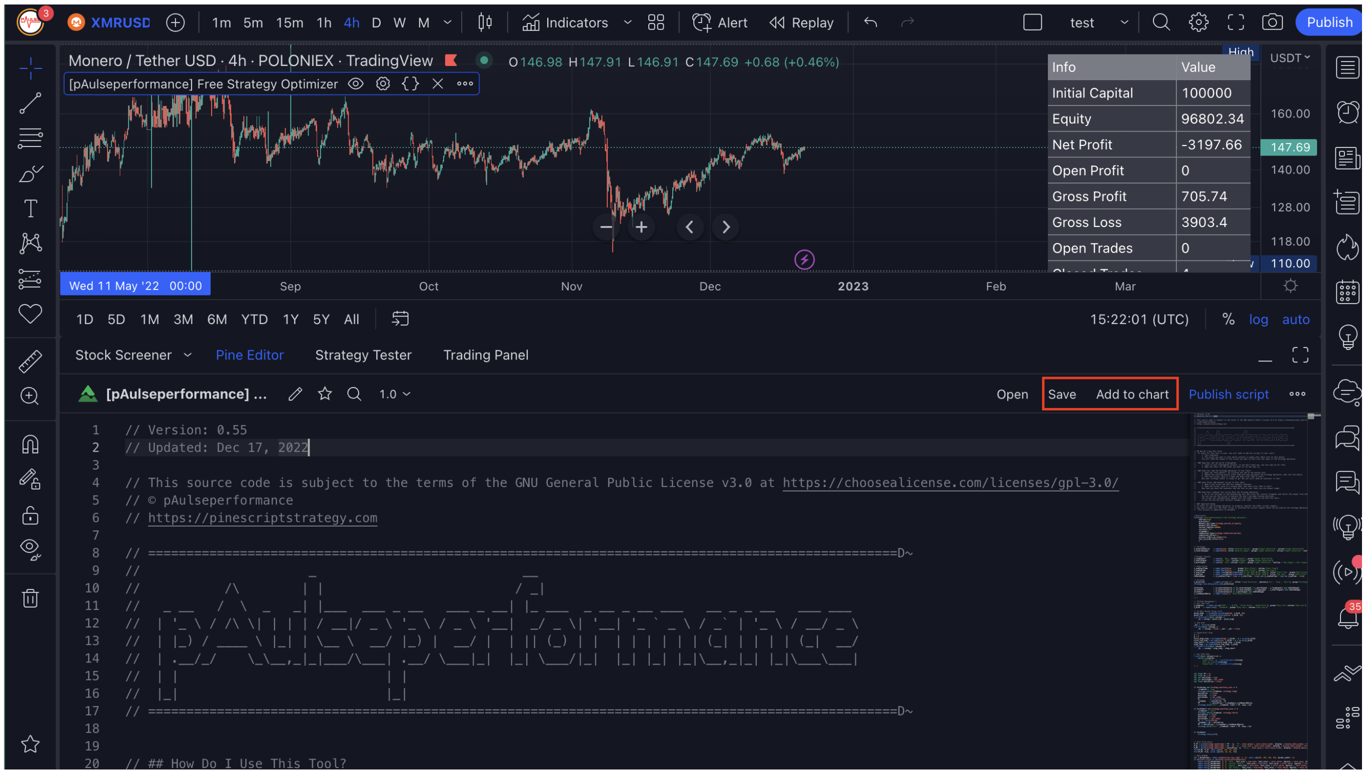
1. Set up an account on TradingView and open the chart for the asset you want to test your strategy on.

2. Add the Strategy Optimizer script to your chart by opening the Pine Editor, copying and pasting the code, and clicking "Add to chart".





Make sure the editor is emtpy before pasting the code.


Monero / Tether USD · 4h · POLONIEX · TradingView O146.98 H147.91 L146.91 C147.69 +0.68 (+0.46%)

[pAulseperformance] Free Strategy Optimizer

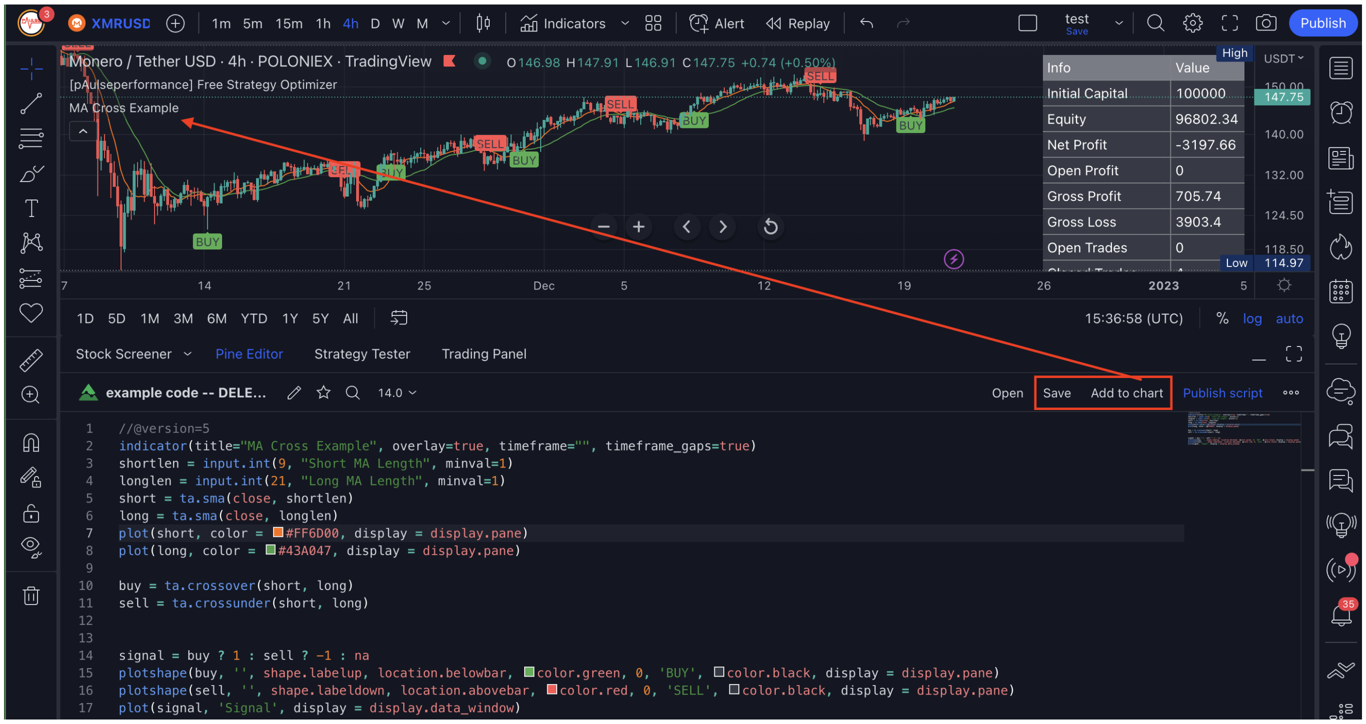
Info	Value
Initial Capital	100000
Equity	96802.34
Net Profit	-3197.66
Open Profit	0
Gross Profit	705.74
Gross Loss	3903.4
Open Trades	0

```
// Version: 0.55
// Updated: Dec 17, 2022

// This source code is subject to the terms of the GNU General Public License v3.0 at https://choosealicense.com/licenses/gpl-3.0/
// © pAulseperformance
// https://pinescriptstrategy.com

// =================================================================================================D~
//                  _                           __
//          /\      | |                         / _|
//  _ __   /  \  _  _| |___  ___ _ __   ___ _ __| |_ ___  _ __ _ __ ___   __ _ _ __   ___ ___
//  | '_ \ / /\ \| | | | / __|/ _ \ '_ \ / _ \ '__|  _/ _ \| '__| '_ ` _ \ / _` | '_ \ / __/ _ \
//  | |_) / ____ \ |_| | \__ \  __/ |_) |  __/ |  | || (_) | |  | | | | | | (_| | | | | (_|  __/
//  | .__/_/    \_\__,_|_|___/\___| .__/ \___|_|  |_| \___/|_|  |_| |_| |_|\__,_|_| |_|\___\___|
//  | |                           | |
//  |_|                           |_|
// =================================================================================================D~


// ## How Do I Use This Tool?
```

3. Add another script to your chart, such as an indicator, by opening a new script, copying and pasting the code below, and clicking "Add to chart".

Use This Code To Get Started

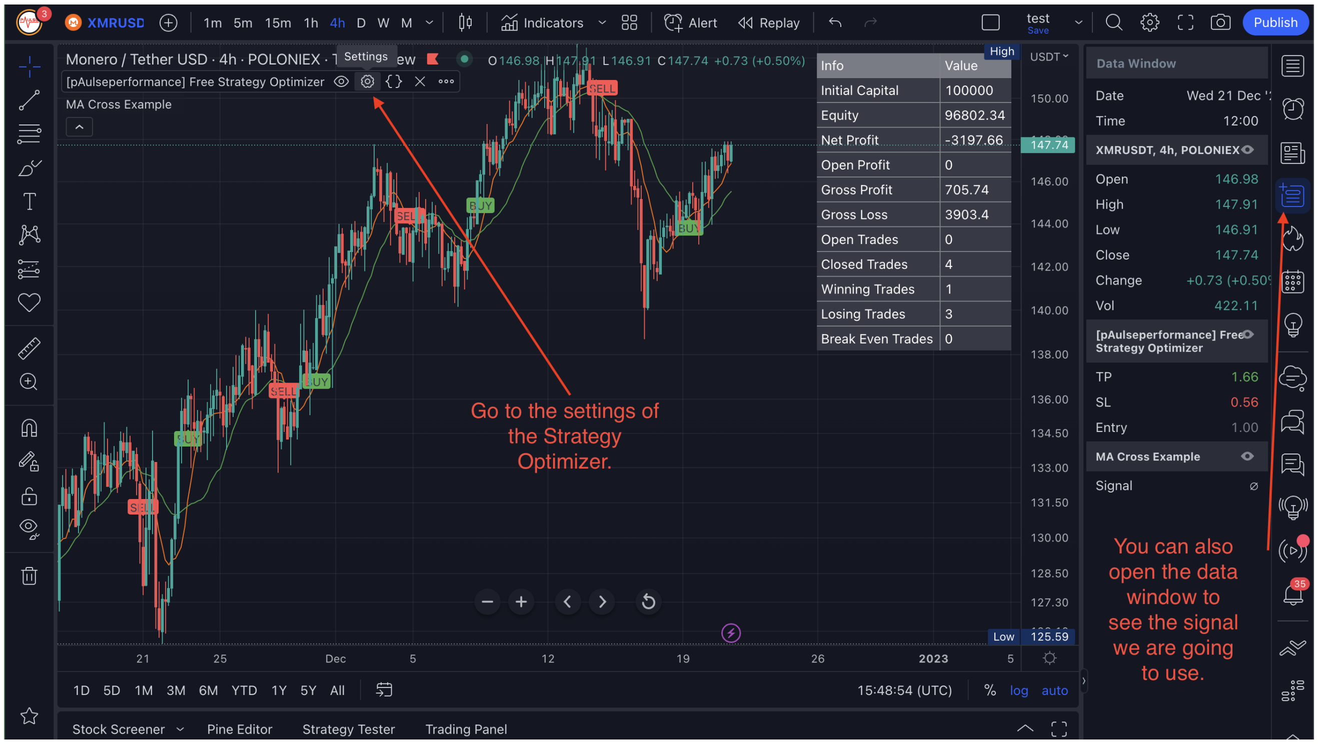
```
//@version=5
indicator(title="MA Cross Example", overlay=true, timeframe="", timeframe_gaps=true)
shortlen = input.int(9, "Short MA Length", minval=1)
longlen = input.int(21, "Long MA Length", minval=1)
short = ta.sma(close, shortlen)
long = ta.sma(close, longlen)
plot(short, color = #FF6D00, display = display.pane)
plot(long, color = #43A047, display = display.pane)
buy = ta.crossover(short, long)
sell = ta.crossunder(short, long)
signal = buy ? 1 : sell ? -1 : na
plotshape(buy, '', shape.labelup, location.belowbar, color.green, 0, 'BUY', color.black, display
= display.pane)
plotshape(sell, '', shape.labeldown, location.abovebar, color.red, 0, 'SELL', color.black,
display = display.pane)
plot(signal, 'Signal', display = display.data_window)
```



Info	Value
Initial Capital	100000
Equity	96802.34
Net Profit	-3197.66
Gross Profit	705.74
Gross Loss	3903.4
Open Profit	0
Open Trades	0

```
//@version=5
indicator(title="MA Cross Example", overlay=true, timeframe="", timeframe_gaps=true)
shortlen = input.int(9, "Short MA Length", minval=1)
longlen = input.int(21, "Long MA Length", minval=1)
short = ta.sma(close, shortlen)
long = ta.sma(close, longlen)
plot(short, color = #FF6D00, display = display.pane)
plot(long, color = #43A047, display = display.pane)

buy = ta.crossover(short, long)
sell = ta.crossunder(short, long)


signal = buy ? 1 : sell ? -1 : na
plotshape(buy, '', shape.labelup, location.belowbar, color.green, 0, 'BUY', color.black, display = display.pane)
plotshape(sell, '', shape.labeldown, location.abovebar, color.red, 0, 'SELL', color.black, display = display.pane)
plot(signal, 'Signal', display = display.data_window)
```

4. In the settings of the Strategy Optimizer, select the output of the other script as the source in the "source" dropdown.

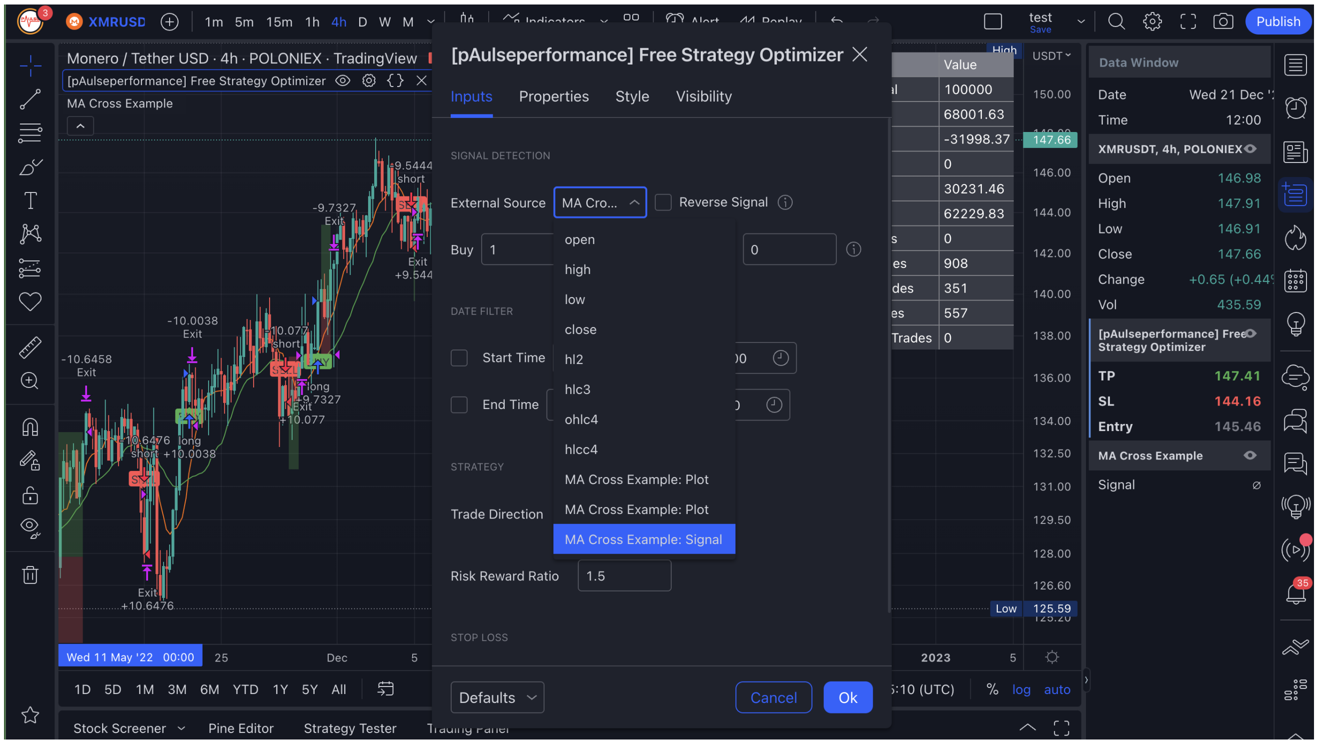

XMRUSD
1m 5m 15m 1h 4h D W M
Indicators
Alert
Replay
Publish
Monero / Tether USD · 4h · POLONIEX
Settings
[pAulseperformance] Free Strategy Optimizer
MA Cross Example
Go to the settings of the Strategy Optimizer.
You can also open the data window to see the signal we are going to use.
Stock Screener
Pine Editor
Strategy Tester
Trading Panel

O146.98 H147.91 L146.91 C147.74 +0.73 (+0.50%)

Info	Value
Initial Capital	100000
Equity	96802.34
Net Profit	-3197.66
Open Profit	0
Gross Profit	705.74
Gross Loss	3903.4
Open Trades	0
Closed Trades	4
Winning Trades	1
Losing Trades	3
Break Even Trades	0

Data Window

Date	Wed 21 Dec '2
Time	12:00
XMRUSDT, 4h, POLONIEX	
Open	146.98
High	147.91
Low	146.91
Close	147.74
Change	+0.73 (+0.50%
Vol	422.11
[pAulseperformance] Free Strategy Optimizer	
TP	1.66
SL	0.56
Entry	1.00
MA Cross Example	
Signal	∅


XMRUSD
1m 5m 15m 1h 4h D W M
Publish
Monero / Tether USD · 4h · POLONIEX · TradingView
[pAulseperformance] Free Strategy Optimizer
MA Cross Example
Stock Screener
Pine Editor
Strategy Tester

[pAulseperformance] Free Strategy Optimizer

Inputs | Properties | Style | Visibility

SIGNAL DETECTION

External Source: MA Cro... ☐ Reverse Signal

Buy: 1 — 0

open
high
low
close
hl2
hlc3
ohlc4
hlcc4
MA Cross Example: Plot
MA Cross Example: Plot
MA Cross Example: Signal

DATE FILTER

☐ Start Time

☐ End Time

STRATEGY

Trade Direction

Risk Reward Ratio: 1.5

STOP LOSS

Defaults | Cancel | Ok

	Value
…l	100000
	68001.63
	-31998.37
	0
	30231.46
	62229.83
…s	0
…es	908
…des	351
…es	557
…Trades	0

Data Window

Date	Wed 21 Dec '2
Time	12:00
XMRUSDT, 4h, POLONIEX	
Open	146.98
High	147.91
Low	146.91
Close	147.66
Change	+0.65 (+0.44
Vol	435.59
[pAulseperformance] Free Strategy Optimizer	
TP	147.41
SL	144.16
Entry	145.46
MA Cross Example	
Signal	∅

✓ **Success**

Now you are ready to use the tool to analyze the chart and make trading decisions. The script will automatically update with the latest data as you move the chart, so you can see how your analysis changes over time.

⚠ **Important**

In order for the Strategy Optimizer to properly register the signals from the other script, it is important to make sure the script is plotting the correct signals that can be used by the Strategy Optimizer. This process requires some manual setup, but we will demonstrate a standard that is easy to follow. Please see the next section for more details.

How Do I Modify Scripts To Work With The Strategy Optimizer?

The Strategy Optimizer can be used to backtest almost any indicator, even closed-source ones (though many developers choose to hide their outputs to avoid this type of testing). The key is to create a single plot in the indicator script that plots an integer output of the trading signals. Here are the steps:

1. Identify the buy/sell signals of the indicator you want to test.
2. Plot the buy/sell signals as integers in a single plot. You can use positive integers for buy signals and negative integers for sell signals. For example:
   - Buy signals: `1`
   - Sell signals: `-1`
   - Exit signals: `0` (You can also add a third output for exit signals, but it is not necessary, as exit signals are usually developed in the Strategy Optimizer.)
3. If you are not familiar with Pine Script, you can ask the developer of the script to do this for you, or you can learn some basics.

It should look something like this when your done.

```
// *Script Code* //

signal = buyCondition ? 1 : sellCondition ? -1 : na
plot(signal, 'Strategy Optimizer Signal')
```

> ***The only thing you have to do, is identify the buy/sell signals of the indicator in question and then plot them in a single plot.***

Let's look at some examples in the next section so you can get a better understanding of how this works.

Modified Script Examples

Example 1: Built-in MA Cross

Here is an example using the built-in MA Cross indicator in Pine Script:

```
//@version=5
indicator(title="MA Cross", overlay=true, timeframe="", timeframe_gaps=true)
shortlen = input.int(9, "Short MA Length", minval=1)
longlen = input.int(21, "Long MA Length", minval=1)
short = ta.sma(close, shortlen)
long = ta.sma(close, longlen)
plot(short, color = #FF6D00)
plot(long, color = #43A047)
plot(ta.cross(short, long) ? short : na, color=#2962FF, style = plot.style_cross, linewidth = 4)
```

In this code, there is slow moving average and a fast moving average.
A cross is plotted every time the moving averages crosses, either up or down.
`plot(ta.cross(short, long) ? short : na, color=#2962FF, style = plot.style_cross, linewidth = 4)`

We need to define clear buy/sell signals where this cross occurs, unfortunately we will have to write a little more code to do this, but still much less than converting this to a strategy.

Here is the modified script ready for the Strategy Optimizer:

```
//@version=5
indicator(title="MA Cross", overlay=true, timeframe="", timeframe_gaps=true)
shortlen = input.int(9, "Short MA Length", minval=1)
longlen = input.int(21, "Long MA Length", minval=1)
short = ta.sma(close, shortlen)
long = ta.sma(close, longlen)
plot(short, color = #FF6D00)
plot(long, color = #43A047)
plot(ta.cross(short, long) ? short : na, color=#2962FF, style = plot.style_cross, linewidth = 4)

buy = ta.crossover(short, long)
sell = ta.crossunder(short, long)
signal = buy ? 1 : sell ? -1 : na

plot(signal, 'Signal', display = display.data_window)
```

We defined the buy and sell signals using the `crossover` and `crossunder` functions, respectively.

We then combined the buy and sell signals into one signal using a ternary operator (`? :`) to return a `1` when the buy signal is triggered, a `-1` when the sell signal is triggered and a `na` (not available) otherwise.

Finally, we plot the `signal` on the chart using the `plot` function.

We also specify a title for the signal (So it's easy to find later), and a display parameter (So it doesn't show up in our chart and make a mess).

Example 2: Built-in Bollinger Bands

This example uses the built-in Bollinger Bands indicator in Pine Script:

```
//@version=5
indicator(shorttitle="BB", title="Bollinger Bands", overlay=true, timeframe="", timeframe_gaps=true)
length = input.int(20, minval=1)
src = input(close, title="Source")
mult = input.float(2.0, minval=0.001, maxval=50, title="StdDev")
basis = ta.sma(src, length)
dev = mult * ta.stdev(src, length)
upper = basis + dev
lower = basis - dev
offset = input.int(0, "Offset", minval = -500, maxval = 500)
plot(basis, "Basis", color=#FF6D00, offset = offset)
p1 = plot(upper, "Upper", color=#2962FF, offset = offset)
p2 = plot(lower, "Lower", color=#2962FF, offset = offset)
fill(p1, p2, title = "Background", color=color.rgb(33, 150, 243, 95))
```

To turn this indicator into a strategy, we need to define clear buy and sell signals based on the Bollinger Bands.

Here is the modified script, ready for the Strategy Optimizer:

```
//@version=5
indicator(shorttitle="BB", title="Bollinger Bands", overlay=true, timeframe="", timeframe_gaps=true)
length = input.int(20, minval=1)
src = input(close, title="Source")
mult = input.float(2.0, minval=0.001, maxval=50, title="StdDev")
```

```
basis = ta.sma(src, length)
dev = mult * ta.stdev(src, length)
upper = basis + dev
lower = basis - dev
offset = input.int(0, "Offset", minval = -500, maxval = 500)
plot(basis, "Basis", color=#FF6D00, offset = offset)
p1 = plot(upper, "Upper", color=#2962FF, offset = offset)
p2 = plot(lower, "Lower", color=#2962FF, offset = offset)
fill(p1, p2, title = "Background", color=color.rgb(33, 150, 243, 95))

// Signal for optimizer
buy = close < lower
sell = close > upper
signal = buy ? 1 : sell ? -1 : na

plot(signal, 'Signal', display = display.data_window)
```

We defined the buy and sell signals using the `<` and `>` operators, respectively. We check if the close price is below the lower Bollinger Band (buy signal) or above the upper Bollinger Band (sell signal).

We then combined the buy and sell signals into one signal using a ternary operator (`? :`) to return a `1` when the buy signal is triggered, a `-1` when the sell signal is triggered and a `na` (not available) otherwise.

Finally, we plot the `signal` on the chart using the `plot` function.

We also specify a title for the signal (So it's easy to find later), and a display parameter (So it doesn't show up in our chart and make a mess).

Example 3: Built-In Williams %R

In the next example, we will use the Strategy Optimizer with a built-in oscillator called the "Williams %R." It is a Wave Oscillator, similar to the RSI.

```
//@version=5
indicator("Williams Percent Range", shorttitle="Williams %R", format=format.price, precision=2,
timeframe="", timeframe_gaps=true)
length = input(title="Length", defval=14)
src = input(close, "Source")
_pr(length) =>
        max = ta.highest(length)
        min = ta.lowest(length)
        100 * (src - max) / (max - min)
percentR = _pr(length)
obPlot = hline(-20, title="Upper Band", color=#0d0e13)
hline(-50, title="Middle Level", linestyle=hline.style_dotted, color=#787B86)
osPlot = hline(-80, title="Lower Band", color=#787B86)
fill(obPlot, osPlot, title="Background", color=color.rgb(126, 87, 194, 90))
plot(percentR, title="%R", color=#7E57C2)
```

To use this script with the Strategy Optimizer, we will need to add buy and sell signals and plot them as a single signal.

```
//@version=5
indicator("Williams Percent Range", shorttitle="Williams %R", format=format.price, precision=2,
timeframe="", timeframe_gaps=true)
length = input(title="Length", defval=14)
src = input(close, "Source")
_pr(length) =>
        max = ta.highest(length)
        min = ta.lowest(length)
```

```
          100 * (src - max) / (max - min)
percentR = _pr(length)
obPlot = hline(-20, title="Upper Band", color=#0d0e13)
hline(-50, title="Middle Level", linestyle=hline.style_dotted, color=#787B86)
osPlot = hline(-80, title="Lower Band", color=#787B86)
fill(obPlot, osPlot, title="Background", color=color.rgb(126, 87, 194, 90))
plot(percentR, title="%R", color=#7E57C2)

buy = percentR < -80
sell = percentR > -20
signal = buy ? 1 : sell ? -1 : na

plot(signal, 'Signal', display = display.data_window)
```

We defined the buy and sell signals using the `>` and `<` operators, respectively.

We then combined the buy and sell signals into one signal using a ternary operator (`? :`) to return a `1` when the buy signal is triggered, a `-1` when the sell signal is triggered and a `na` (not available) otherwise.

Finally, we plot the `signal` on the chart using the `plot` function.

We also specify a title for the signal (So it's easy to find later), and a display parameter (So it doesn't show up in our chart and make a mess).

Example 4: Custom Squeeze Momentum Indicator [LazyBear]

This is the top indicator on TV as of December 2022, based on likes.

It's possible that its popularity is due to its history as one of the first scripts published on TradingView, rather than its effectiveness as an indicator. However, we won't know for sure until we put it to the test.
Regardless of the reason, this indicator is a classic, and currently uses Version 1 of Pine Script, while we are now on Version 5.

Normally, if you wanted to convert this to a strategy the traditional way, you would have to convert it to version 5 first and then add your strategy logic.

Fortunately, the Strategy Optimizer allows us to use this script without worrying about compatibility issues, as long as we make some minor adjustments.

Here's the original code

```
//
// @author LazyBear
//

study(shorttitle = "SQZMOM_LB", title="Squeeze Momentum Indicator [LazyBear]", overlay=false)
length = input(20, title="BB Length")
mult = input(2.0,title="BB MultFactor")
lengthKC=input(20, title="KC Length")
multKC = input(1.5, title="KC MultFactor")

useTrueRange = input(true, title="Use TrueRange (KC)", type=bool)

// Calculate BB
source = close
basis = sma(source, length)
dev = multKC * stdev(source, length)
upperBB = basis + dev
lowerBB = basis - dev
```

```
// Calculate KC
ma = sma(source, lengthKC)
range = useTrueRange ? tr : (high - low)
rangema = sma(range, lengthKC)
upperKC = ma + rangema * multKC
lowerKC = ma - rangema * multKC

sqzOn = (lowerBB > lowerKC) and (upperBB < upperKC)
sqzOff = (lowerBB < lowerKC) and (upperBB > upperKC)
noSqz = (sqzOn == false) and (sqzOff == false)

val = linreg(source - avg(avg(highest(high, lengthKC),
lowest(low,lengthKC)),sma(close,lengthKC)),lengthKC,0)

bcolor = iff( val > 0,iff( val > nz(val[1]), lime, green),iff( val < nz(val[1]), red, maroon))
scolor = noSqz ? blue : sqzOn ? black : gray
plot(val, color=bcolor, style=histogram, linewidth=4)
plot(0, color=scolor, style=cross, linewidth=2)
```

To use this script with the Strategy Optimizer, we will need to add buy and sell signals and plot them as a single signal.

```
//
// @author LazyBear
//

study(shorttitle = "SQZMOM_LB", title="Squeeze Momentum Indicator [LazyBear]", overlay=false)
length = input(20, title="BB Length")
mult = input(2.0,title="BB MultFactor")
lengthKC=input(20, title="KC Length")
multKC = input(1.5, title="KC MultFactor")

useTrueRange = input(true, title="Use TrueRange (KC)", type=bool)

// Calculate BB
source = close
basis = sma(source, length)
dev = multKC * stdev(source, length)
upperBB = basis + dev
lowerBB = basis - dev

// Calculate KC
ma = sma(source, lengthKC)
range = useTrueRange ? tr : (high - low)
rangema = sma(range, lengthKC)
upperKC = ma + rangema * multKC
lowerKC = ma - rangema * multKC

sqzOn = (lowerBB > lowerKC) and (upperBB < upperKC)
sqzOff = (lowerBB < lowerKC) and (upperBB > upperKC)
noSqz = (sqzOn == false) and (sqzOff == false)

val = linreg(source - avg(avg(highest(high, lengthKC),
lowest(low,lengthKC)),sma(close,lengthKC)),lengthKC,0)

bcolor = iff( val > 0,iff( val > nz(val[1]), lime, green),iff( val < nz(val[1]), red, maroon))
scolor = noSqz ? blue : sqzOn ? black : gray
plot(val, color=bcolor, style=histogram, linewidth=4)
plot(0, color=scolor, style=cross, linewidth=2)

// Added for strategy optimizer
```

```
buy = sqzOn[1] and sqzOff and val > 0
sell = sqzOn[1] and sqzOff and val < 0
exit = noSqz
signal = buy ? 1 : sell ? -1 : exit ? 0: na

plot(signal, 'Signal')
```

We defined the buy and sell signals using logic from the indicator visual cues.

A `buy` happens when the indicator transitions from `black` to `gray` crosses and the oscillator is `green`
`buy = sqzOn[1] and sqzOff and val > 0`
A `sell` happens similarly, except we wait for the oscillator to be `red.`
`sell = sqzOn[1] and sqzOff and val < 0`
This way we capitalize on a squeeze in any direction.

We've also include and `exit` symbol.
`exit = noSqz`

We then combined the buy, sell and exit signals into one signal using a ternary operator (`? :`) to return a `1` when the buy signal is triggered, a `-1` when the sell signal is triggered, a `0` when the exit signal is triggered and a `na` (not available) otherwise.

Finally, we plot the `signal` on the chart using the `plot` function.

We also specify a title for the signal (So it's easy to find later); however, we were unable to hide the signal from the chart, because that feature is not available in version 1.

Testing Closed Source Scripts

A Warning On Closed Source Scripts For Sale

It is important to be aware that not all indicators and strategies available on the market are of high quality. Some vendors may sell indicators that do not perform as advertised or that are not suitable for certain market conditions. In some cases, the only people who benefit from these indicators are the vendors themselves, as they are able to sell them to unsuspecting traders who may not realize that they are not effective.

To avoid falling victim to such scams, it is important to do your due diligence and thoroughly research any indicator or strategy before purchasing it. Look for reviews and testimonials from real users, and consider trying out a free or trial version of the tool to see if it meets your expectations. It is also a good idea to use caution when dealing with vendors who make overly-optimistic or unrealistic claims about their products. Remember that no indicator or strategy is guaranteed to be profitable, and it is always important to manage your risk carefully and use good judgment when making trading decisions.

Note from the author

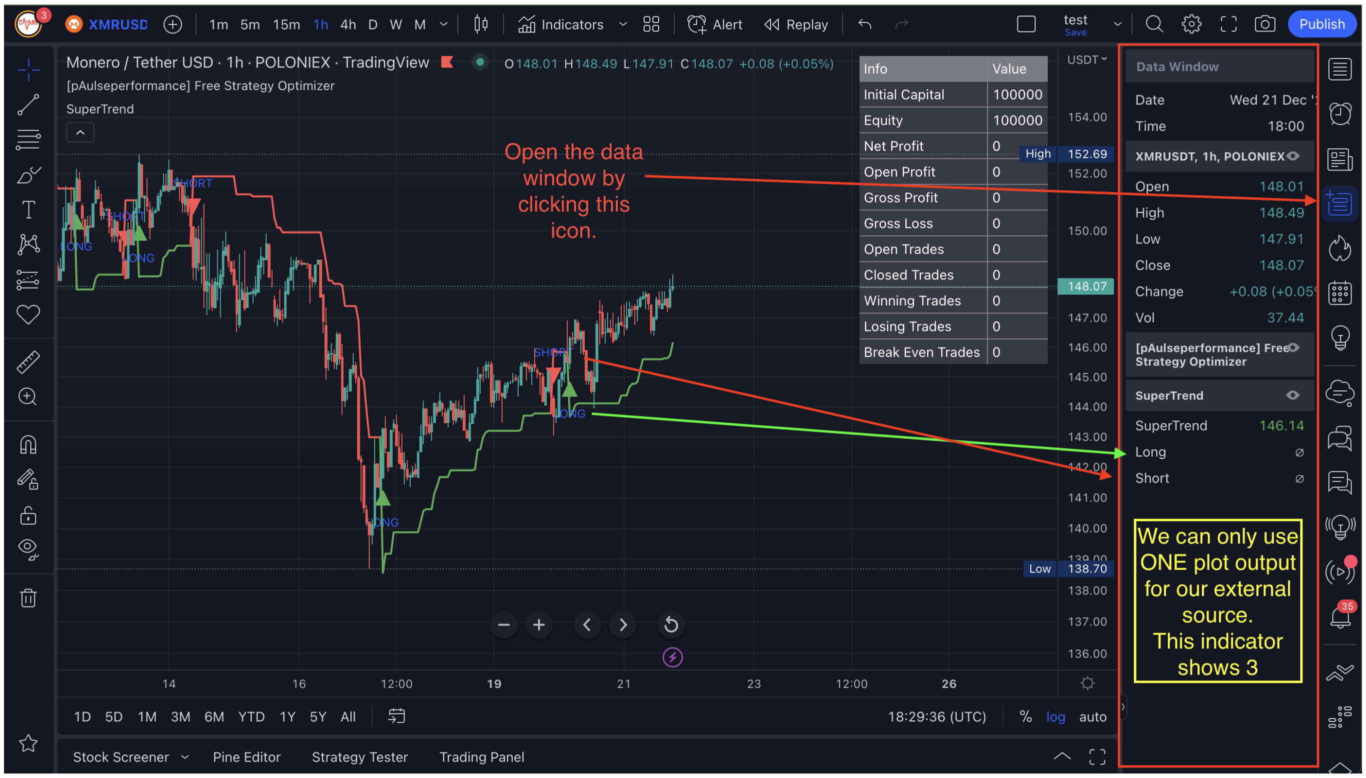
I've been writing Pine since 2015 and TradingView has come a long way, but one thing they need to fix is close source indicator vendors.

In my opinion, all of the paid for indicators sold by vendors are not worth a penny because of one simple fact. You cannot backtest them.

This tool helps with that, but due to the hidden code and smart developers, I often find that vendors smarten up and hide their plot outputs so they cannot be backtested.
If you have any doubts about the fecundity of the indicator, ask the vendor to supply you with a buy/sell signal output similar to the one we are using. You can even show them this tool as an example. If they are confident in the results of their indicator, they should be ok with this.

If they object, I highly advise reconsidering your purchase, unless they can provide an alternative to real verified trading results.

Since we don't have access to the code of close source scripts, we need to look at the Data Window inside TradingView and determine if there is a signal we can feed into our Strategy Optimizer.


XMRUSD
1m 5m 15m 1h 4h D W M
Indicators
Alert
Replay
test
Save
Publish
Open the data window by clicking this icon.
We can only use ONE plot output for our external source. This indicator shows 3
SHORT
LONG
1D 5D 1M 3M 6M YTD 1Y 5Y All
18:29:36 (UTC)
% log auto
Stock Screener
Pine Editor
Strategy Tester
Trading Panel

Monero / Tether USD · 1h · POLONIEX · TradingView O148.01 H148.49 L147.91 C148.07 +0.08 (+0.05%)
[pAulseperformance] Free Strategy Optimizer
SuperTrend

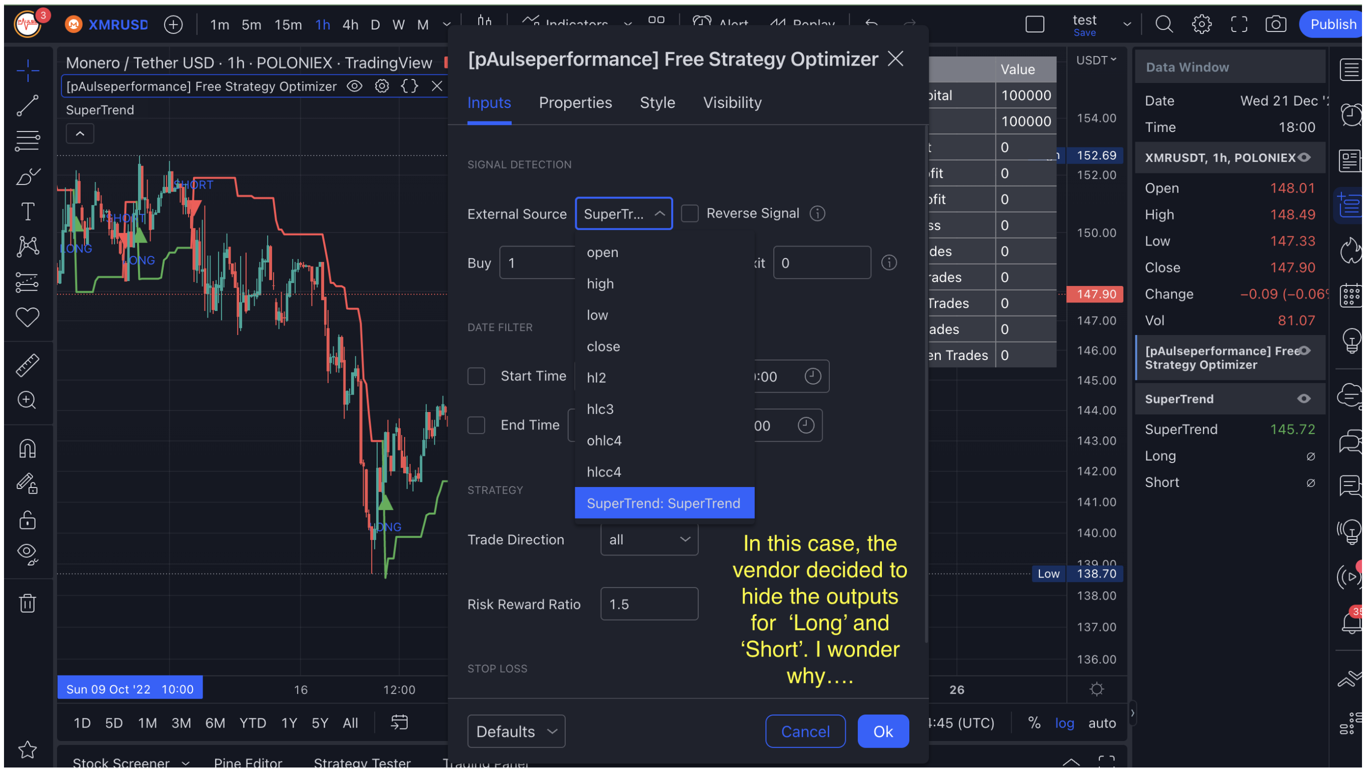
Info	Value
Initial Capital	100000
Equity	100000
Net Profit	0
Open Profit	0
Gross Profit	0
Gross Loss	0
Open Trades	0
Closed Trades	0
Winning Trades	0
Losing Trades	0
Break Even Trades	0

Data Window

Date	Wed 21 Dec '
Time	18:00
XMRUSDT, 1h, POLONIEX	
Open	148.01
High	148.49
Low	147.91
Close	148.07
Change	+0.08 (+0.05
Vol	37.44
[pAulseperformance] Free Strategy Optimizer	
SuperTrend	
SuperTrend	146.14
Long	∅
Short	∅

I've added a random Custom Closed Source indicator I found. You can add it here

It plots Long and Short signals. We would love to test these. We can see them in the data window, but they might not still be accessible in the Strategy Optimizer if the script author hides them further.



[pAulseperformance] Free Strategy Optimizer

Inputs Properties Style Visibility

SIGNAL DETECTION

External Source: SuperTr...
Reverse Signal

Buy: 1

Exit: 0

open
high
low
close
hl2
hlc3
ohlc4
hlcc4
SuperTrend: SuperTrend

DATE FILTER

Start Time :00
End Time 00

STRATEGY

Trade Direction: all

Risk Reward Ratio: 1.5

STOP LOSS

Defaults Cancel Ok

In this case, the vendor decided to hide the outputs for 'Long' and 'Short'. I wonder why….

	Value
pital	100000
	100000
t	0
fit	0
ofit	0
ss	0
des	0
rades	0
Trades	0
ades	0
en Trades	0

Data Window

Date	Wed 21 Dec '
Time	18:00
XMRUSDT, 1h, POLONIEX	
Open	148.01
High	148.49
Low	147.33
Close	147.90
Change	−0.09 (−0.06
Vol	81.07
[pAulseperformance] Free Strategy Optimizer	
SuperTrend	
SuperTrend	145.72
Long	∅
Short	∅

In this case, the author did decide to hide the `Long` and `Short` signals. If they were available, we would test the long and short signals separately, but we can't even do that!

We still have the `Supertrend` output that we can use here. In this case, we could modify our Strategy optimizer to reverse engineer the signals based off this indicator, but thats out of the scope of this book.

What Is The Best Trading Strategy?

There is no one-size-fits-all best trading strategy. What works for one person may not work for another, and what works in one market may not work in another. The best trading strategy is one that takes into account your individual circumstances, such as your risk tolerance and investment goals, and is based on sound principles of risk management.

If you'd like to learn more about building your own indicators and trading strategies, check out these courses

Strategy Optimizer Pro

If you like this tool and wish to have more features, check out the Strategy Optimizer Pro.