Pair Strength Analyzer (PSA)

This is an indicator that I used as an adjunct to my strength indicators, to confirm strength/weakness by using a very different algorithm. I've finally decided to share it publicly. There is nothing magical about it; it is merely another way of calculating trend strength, and — as always — the question is whether the trend (on your chosen timeframe/horizon) remains effective, on average, for long enough for you to profit from it. The indicator ranks currencies, from the strongest trending down to the most sideways moving, according to the parameter settings that you decide to apply.

The indicator was written and compiled using MT4 build 509. I have no idea whether it will work correctly on build 600 or later.

TERMS AND CONDITIONS

Please note that all code in this thread is supplied FREE of charge. Therefore the following conditions apply:

1. You agree that, if you download and use the code, it is **ENTIRELY AT YOUR OWN RISK**. I accept **NO LIABILITY** for any financial losses or computer related damage, caused by either the correct or incorrect use of the code.

2. Feel welcome to share the code freely, and modify any MQ4 source. However, you may **NOT** sell, or otherwise distribute, **any part of the code** commercially, without my prior written consent.

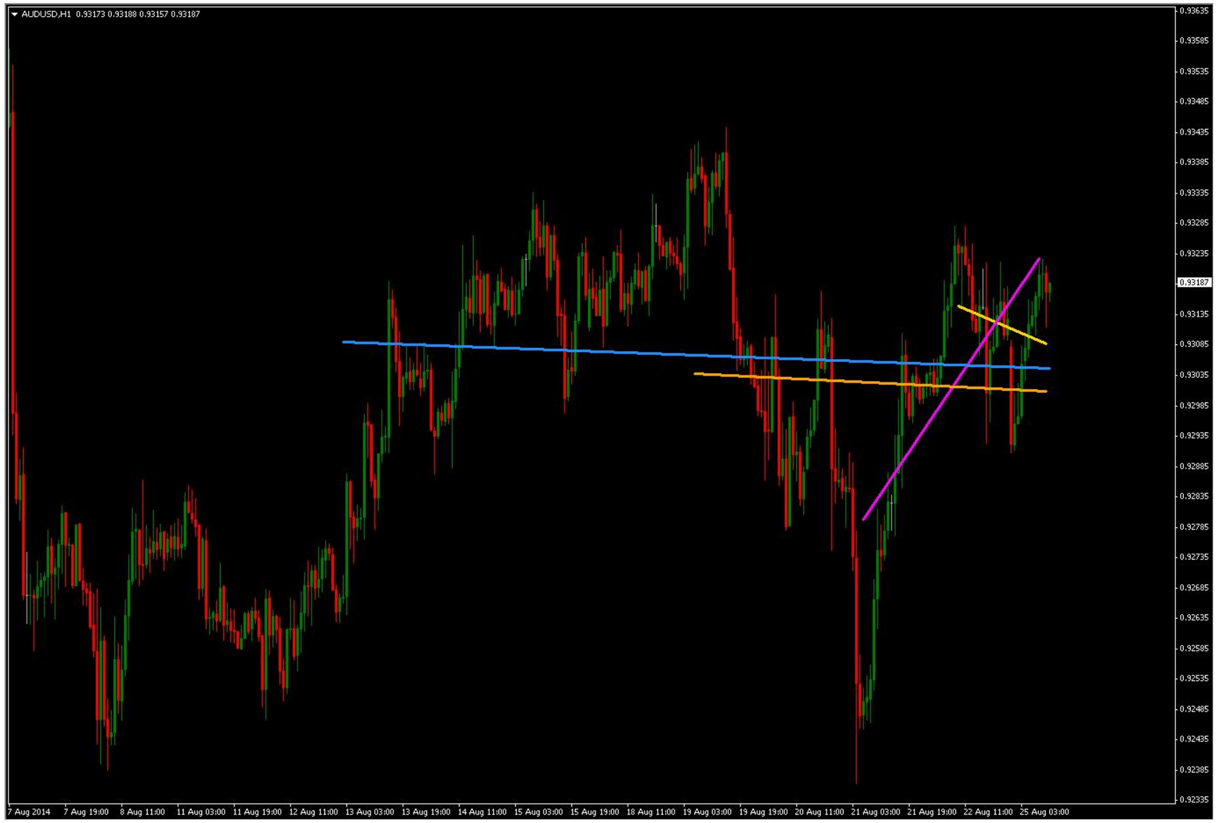
3. The code may not run correctly on Windows 7, Windows 8 or Vista, probably for the reasons given here: http://4xtrader.net/how-to-run-metatrader-on-windows-7-or-vista/

4. I'm sorry, but due to my current work commitments, **I am no longer modifying code to suit people's personal requirements, nor posting replies to individual questions in this thread**. (If you can't get the code to work, you may find solutions already posted somewhere in the thread; otherwise, you'll need to find another indicator).

METHODOLOGY

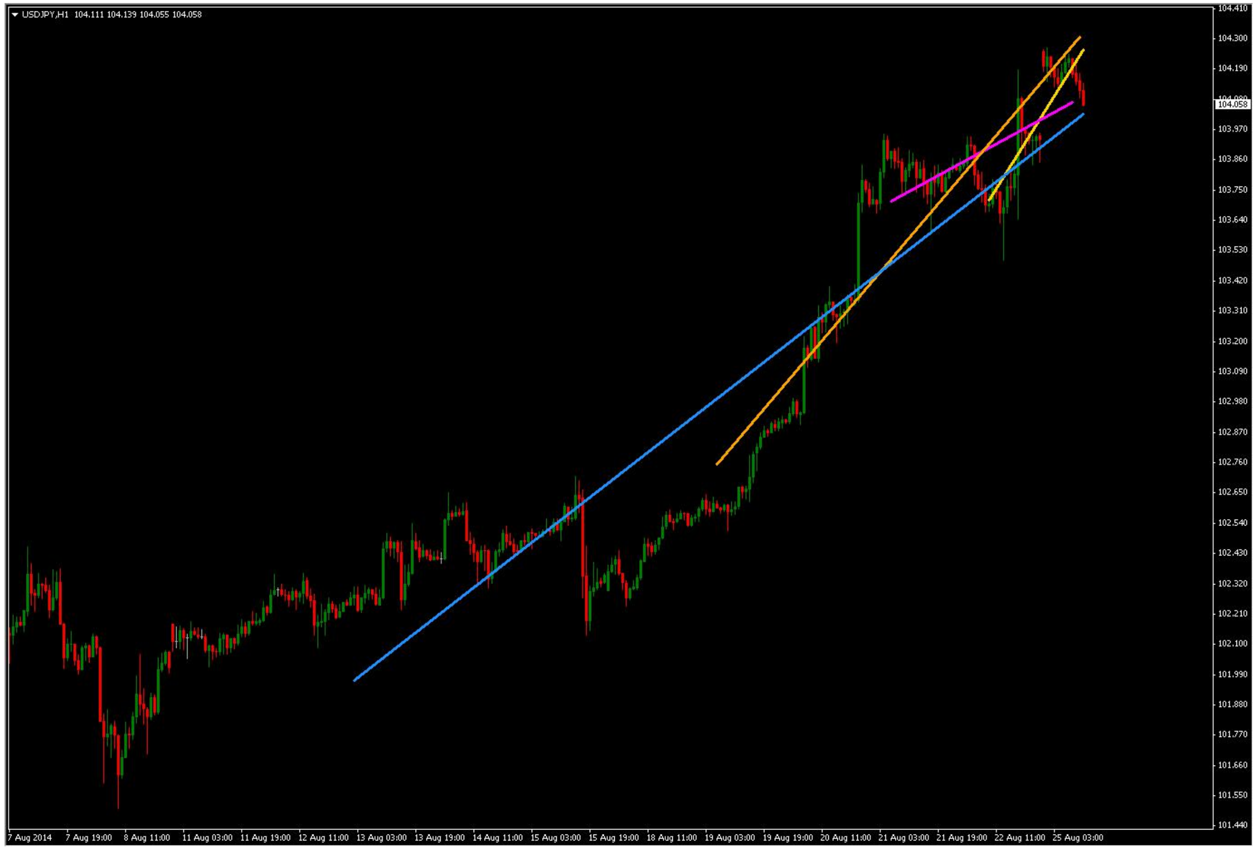
PSA determines trend strength by calculating a weighted average of the slopes of a number of (invisible) regression lines. In each of the screenshots below, I have manually plotted 4 regression lines of different lengths: 25 candle (yellow), 50 candle (magenta), 100 candle (orange) and 200 candle (blue).

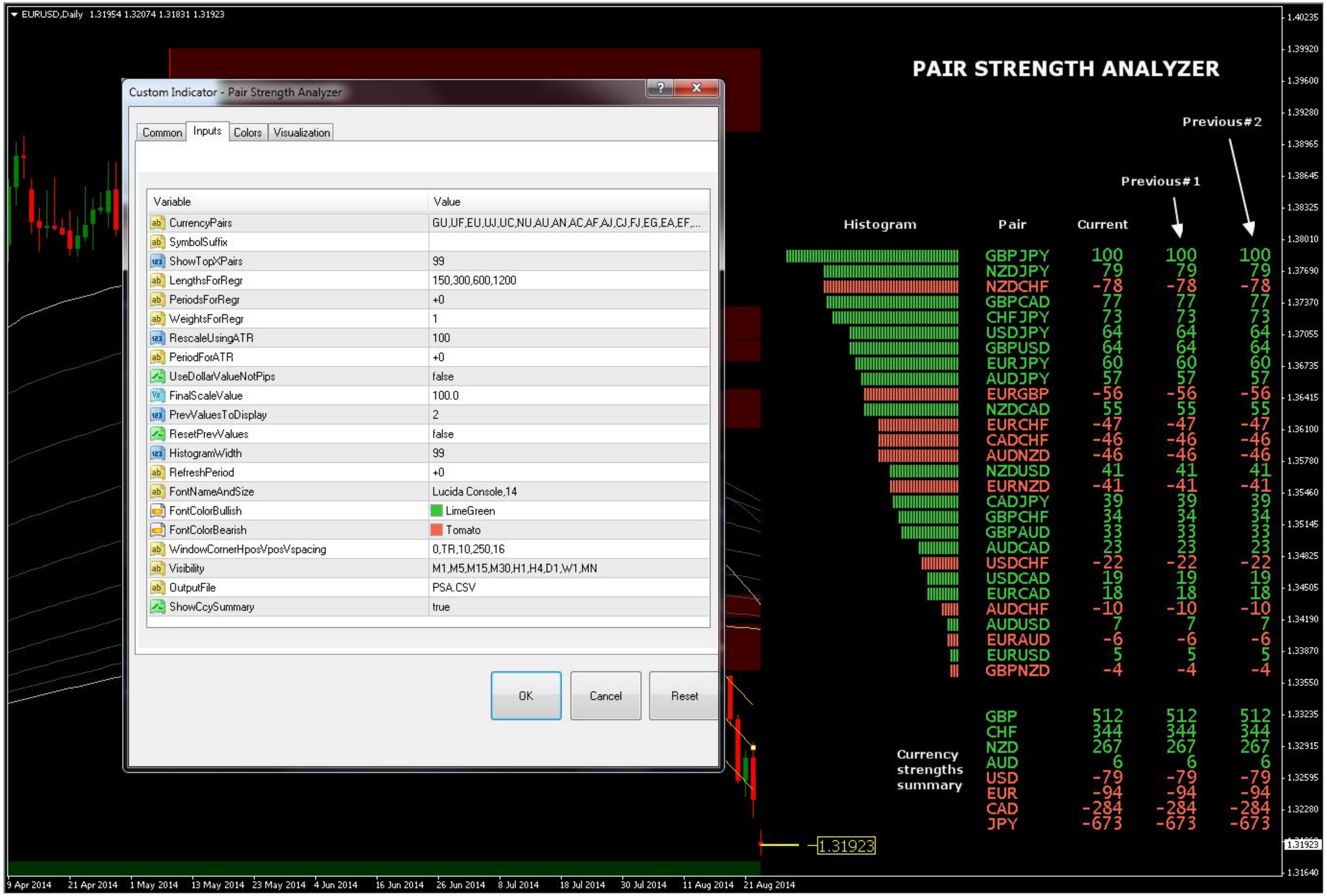
In the following screenshot (AUDUSD, H1), both AUD and USD are strong, i.e. positively correlated, resulting in a sideways or ‘trendless’ chart. The directions/slopes of the 4 regression lines are conflicting and scattered, and averaging their slopes will result in a low summary value (close to zero), as the directions cancel each other.



In the next screenshot (USDJPY,H1), USD is strong and JPY weak, i.e. negative correlation, resulting in a strongly uptrending chart. The directions/slopes of the 4 regression lines are all pointing upward, and averaging their slopes will result in a high positive summary value, as the directions reinforce each other.

(Conversely, in a strongly downtrending chart, the directions of the regression lines would all point downward, and averaging their slopes would result in a high negative summary value).





Variable	Value
CurrencyPairs	GU,UF,EU,UJ,UC,NU,AU,AN,AC,AF,AJ,CJ,FJ,EG,EA,EF,...
SymbolSuffix	
ShowTopXPairs	99
LengthsForRegr	150,300,600,1200
PeriodsForRegr	+0
WeightsForRegr	1
RescaleUsingATR	100
PeriodForATR	+0
UseDollarValueNotPips	false
FinalScaleValue	100.0
PrevValuesToDisplay	2
ResetPrevValues	false
HistogramWidth	99
RefreshPeriod	+0
FontNameAndSize	Lucida Console,14
FontColorBullish	🟩 LimeGreen
FontColorBearish	🟥 Tomato
WindowCornerHposVposVspacing	0,TR,10,250,16
Visibility	M1,M5,M15,M30,H1,H4,D1,W1,MN
OutputFile	PSA.CSV
ShowCcySummary	true

PAIR STRENGTH ANALYZER

Previous#2

Previous#1

Pair	Current	Previous#1	Previous#2
GBPJPY	100	100	100
NZDJPY	79	79	79
NZDCHF	-78	-78	-78
GBPCAD	77	77	77
CHFJPY	73	73	73
USDJPY	64	64	64
GBPUSD	64	64	64
EURJPY	60	60	60
AUDJPY	57	57	57
EURGBP	-56	-56	-56
NZDCAD	55	55	55
EURCHF	-47	-47	-47
CADCHF	-46	-46	-46
AUDNZD	-46	-46	-46
NZDUSD	41	41	41
EURNZD	-41	-41	-41
CADJPY	39	39	39
GBPCHF	34	34	34
GBPAUD	33	33	33
AUDCAD	23	23	23
USDCHF	-22	-22	-22
USDCAD	19	19	19
EURCAD	18	18	18
AUDCHF	-10	-10	-10
AUDUSD	7	7	7
EURAUD	-6	-6	-6
EURUSD	5	5	5
GBPNZD	-4	-4	-4

Currency strengths summary

GBP	512	512	512
CHF	344	344	344
NZD	267	267	267
AUD	6	6	6
USD	-79	-79	-79
EUR	-94	-94	-94
CAD	-284	-284	-284
JPY	-673	-673	-673

PARAMETERS

```
extern string  CurrencyPairs              =
"GU,UF,EU,UJ,UC,NU,AU,AN,AC,AF,AJ,CJ,FJ,EG,EA,EF,EJ,EN,EC,GF,GA,GC,GJ,GN,NJ,NC,CF,NF";
```

Enter up to 99 currency pair symbols, or abbreviations, separated by commas. In typing the symbol, upper/lowercase may be used interchangeably (e.g. USD or usd). Permissible abbreviations are: A=AUD; C=CAD; E=EUR; F=CHF; G=GBP; J=JPY; N=NZD; U=USD; H=HKD; S=SGD; Z=ZAR. So you could type G or g instead of GBP, for example. If the currency name typed is not exactly 1 character, the abbreviation will not be recognized.
You can include metals like XAUUSD or XAGUSD in the list. However, depending on how your br0ker prices these, you may need to use the *RescaleUsingATR* parameter to standardize the result with the currencies.

```
extern string  SymbolSuffix               = "";
```

Use this if your br0ker uses symbols like USDJPYm (in which case you would type m here). The suffix you enter will be appended to all symbols you entered in *CurrencyPairs*. Otherwise simply leave the setting blank.

```
extern int     ShowTopXPairs              = 99;
```

This is the number of strongest trending pairs that you want to be displayed. Leaving the default of 99 means that all pairs will be displayed.

```
extern string  LengthsForRegr             = "5,10,20,50,125,250,500";
```

These are the lengths (number of candles, counting back from the rightmost candle) of the regression lines whose slopes will be averaged. You may make up to 50 entries here, separated by commas. The default setting means that the average slope of 7 regression lines will be used, of length 5,10,20,50,125,250,500. If you wish to add weight to a particular length, type its number multiple times (e.g. typing it twice gives it double the weight, three times triple the weight, and so on).

By having several different regressions of similar length (e.g. 5,10,15,20,25,30,35,40), the summary value displayed by the indicator will factor in the smoothness (consistency), as well as the overall strength, of the trend.

```
extern string  PeriodsForRegr             = "+0";
```

You can also have the regression lines repeat themselves over multiple timeframes. You can either enter timeframes as relative to the current chart, separated by commas, i.e. +0 is the current timeframe (**note: be sure to type the + sign**), +1 is the next longer timeframe, +2 is the next longer timeframe after that, etc; –1 is the next shorter timeframe, –2 is the next shorter timeframe, etc. For example, if you’re on a H1 chart, and you enter –1,+0,+1, then the slope of 21 regression lines would be averaged, i.e. lines of length 5,10,20,50,125,250,500 on each of the M30, H1 and H4 timeframes. But then if you were to switch to the M15 chart, the slopes of lines of length 5,10,20,50,125,250,500 on each of the M5, M15 and M30 timeframes would all be averaged.

Alternatively, you can enter absolute timeframes like H1, H4, D1 (upper or lowercase). Then these timeframes will used no matter what the timeframe of the currently displayed chart is, i.e. they remain ‘locked’ if you switch timeframes.

```
extern string  WeightsForRegr              = "1";
```

This allows you to specify different weights for each of the corresponding timeframes in *PeriodsForRegr*. For example, if *PeriodsForRegr* is H1, H4 and *WeightsForRegr* is 3,1 then the H1 regression slopes will be weighted 3 times as highly as the H4 regression slopes, when the overall averaging is performed.

The *PeriodsForRegr* and *WeightsForRegr* allow you to create a summary of each pair’s strength, into a single value, weighted over any or all of the 9 timeframes available in MT4 (M1,M5,M15,M30,H1,H4,D1,W1,MN), however you wish.

```
extern int     RescaleUsingATR             = 100;
```

You can re-scale all of the averaged values by entering a positive (>0) here, which will be the ATR (MT4’s built-in average true range function) used. For example, if you use the default value of 100, then each slope value will be divided by a 100 period ATR, before they are summed and averaged. This ensures standardization of different units, e.g. pips, points, across different instrument types (currencies, metals, etc).
If you enter 0 (or a negative value), then no re-scaling will be performed, i.e. the ‘raw’ value of each slope will be used in the averaging process.

```
extern string  PeriodForATR                = "+0";
```

This is the timeframe that will be used in conjunction with the *RescaleUsingATR* process. You can enter either a relative or absolute value, exactly for PeriodsForRegr. Hence if you were to enter D1 here, and set *RescaleUsingATR* to 20, then each slope value would be divided by the pair’s 20 day ATR before being passed to the averaging process.
If *RescaleUsingATR* is set to 0, then this parameter is ignored.

```
extern bool    UseDollarValueNotPips       = false;
```

If set to TRUE, this will cause all of the slope values to be multiplied by the ‘dollar per pip’ (‘tickvalue’ in MT4 terminology) before the averaging process takes place. Hence the end result will compare dollar values rather than pips. To make this meaningful, this assumes that you have set *RescaleUsingATR* to 0.
If set to FALSE, the raw pip based slop values are passed on to the averaging process.

```
extern double  FinalScaleValue             = 100;
```

After the weighted averages (summary values) for each pair have been calculated, you can further have these scaled so that the strongest trending pair has a value of 100, and then all other pair’s summary values are ratioed down accordingly. Arguably keeps things neat, especially if you are making comparisons of different instances of the indicator with different regression settings.

```
extern int     PrevValuesToDisplay         = 0;
extern bool    ResetPrevValues            = false;
```

In addition to the indicator plotting the current summary values, you can also have the indicator plot up to 2 prior historical values (the summary values are recalculated and redisplayed periodically, according to the *RefreshPeriod* setting). *PrevValuesToDisplay* may have a value of 0, 1 or 2, depending on how many prior values you wish to display, each in a separate column. For example, if the *RefreshPeriod* is M30, then every 30 minutes the previous#2 value disappears, the previous#1 value becomes the new previous#2 value, the current value becomes the new previous#1 value, and a new current value is calculated, as the new M30 candle forms. The current and prior values are all stored in Global Variables, which means that they should remain 'sticky' if MT4 is re-started. If you want to reset all prior values to their current values, set *ResetPrevValues* to TRUE.

New feature: you can set *PrevValuesToDisplay* to –1 or –2. This likewise displays 1 or 2 prior historical values, but the values displayed are the increase (positive value) or decrease (negative value) from the previous value, as opposed to the previous value itself.

```
extern string  HistWidthSymbolSizeOffset    = "0,167,10,140";
```

The indicator can also plot a histogram of the summary values. The 4 integer values, separated by commas, are (from left to right):

1. Enter a value between 10 and 60 to set the maximum width (number of bars). The largest value (at the top of the list of pairs) will plot this number of bars, and all lower values will be scaled accordingly. To disable the plotting of the histogram, set this to 0 (or simply blank out the whole parameter).
2. The value of the Wingdings symbol code you want plotted, for each histogram bar. Refer to the Wingdings symbol chart below, and enter its numerical value.
3. The size of the Wingdings font to be used.
4. Sets the number of pixels away from the pair id that the first histogram bar will display. Increasing this value will create a larger gap between the pair id and the histogram.

```
extern string  RefreshPeriod               = "+0";
```

This controls how often the summary values are recalculated and redisplayed. You can enter either a relative or absolute timeframe, as previously explained in *PeriodsForRegr*.

```
extern string  FontNameAndSize             = "Lucida Console,14";
```

Type the name of the font (you must get the spelling exactly correct, for Windows to recognize the font) you want to use, in the displaying of the summary values; then a comma; then the font size you wish to be used.

```
extern color   FontColorBullish            = LimeGreen;
```

If the average slope is greater than 0, then the currency is deemed to be in an uptrend, and the positive summary value will be displayed using the color you select here. Trend traders should be looking to place buy orders, according to their entry rules.

```
extern color   FontColorBearish              = Tomato;
```

If the average slope is less than 0, then the currency is deemed to be in an downtrend, and the negative summary value will be displayed using the color you select here. Trend traders should be looking to place sell orders, according to their entry rules.

```
extern string  WindowCornerHposVposVspacing = "0,TR,10,250,16";
```

These entries control where the summary table will be displayed on the chart, and its format. You must enter 5 values, separated by commas.
First value is the (sub)window in which you want the values to be displayed. 0 is the main price chart, 1 is the first sub-window below that, 2 is the next sub-window, etc.
Second value is the corner: enter either TL (top left), TR (top right), BL (bottom left), BR (bottom right).
Third value is the starting horizontal pixel number.
Fourth value is the starting vertical pixel number.
Fifth value is the vertical spacing (number of pixels) between each displayed entry (instrument). With a font size of 14, a spacing of 16 allows a small gap (16–14=2) between each row.

```
extern string  Visibility                   = "M1,M5,M15,M30,H1,H4,D1,W1,MN";
```

This controls which timeframe charts the summary will be displayed on. Enter valid values (M1, M5, etc) separated by commas.

```
extern string  OutputFile                   = "PSA.CSV";
```

The indicator can output the same summary that’s displayed on screen to a text file. The file could then be read by an EA, and the EA make decisions on which pairs to trade, based on the pair strength values in the file.
If you enter any non-blank text, that will be the name of the file that will be created in MT4’s *Files* folder. The file gets overwritten with new values, every time the on-screen values are updated, which is determined by the *RefreshPeriod*.
If you leave the parameter setting blank, then no file will be created.

```
extern bool    ShowCcySummary               = true;
```

The indicator can also produce a summary of the currencies, which it does by averaging all of the summary values displayed for the pairs. Set this parameter to TRUE to have this summary displayed, or FALSE to disable it.

WINGDINGS SYMBOL CHART — see next page

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	32	33 🖉	34 ✂	35 ✁	36 👓	37 🕭	38 🕮	39 🕯	40 🕿	41 ✆	42 🖂	43 🖃	44 📪	45 📫	46 📬	47 📭
30	48 📁	49 📂	50 📄	51 🗏	52 🗐	53 🗄	54 ⌛	55 🖮	56 🖰	57 🖲	58 🖳	59 🖴	60 🖫	61 🖬	62 ✇	63 ✍
40	64 🖎	65 ✌	66 👌	67 👍	68 👎	69 ☜	70 ☞	71 ☝	72 ☟	73 🖐	74 ☺	75 😐	76 ☹	77 💣	78 ☠	79 🏳
50	80 🏱	81 ✈	82 ☼	83 💧	84 ❄	85 🕆	86 ✞	87 🕈	88 ✠	89 ✡	90 ☪	91 ☯	92 ॐ	93 ☸	94 ♈	95 ♉
60	96 ♊	97 ♋	98 ♌	99 ♍	100 ♎	101 ♏	102 ♐	103 ♑	104 ♒	105 ♓	106 🙰	107 🙵	108 ●	109 🔾	110 ■	111 □
70	112 🞐	113 ❑	114 ❒	115 ⬧	116 ⧫	117 ◆	118 ❖	119 ⬥	120 ⌧	121 ⮹	122 ⌘	123 🏵	124 🏶	125 🙶	126 🙷	127 ▯
80	128 ⓪	129 ①	130 ②	131 ③	132 ④	133 ⑤	134 ⑥	135 ⑦	136 ⑧	137 ⑨	138 ⑩	139 ⓿	140 ❶	141 ❷	142 ❸	143 ❹
90	144 ❺	145 ❻	146 ❼	147 ❽	148 ❾	149 ❿	150 🙢	151 🙠	152 🙡	153 🙣	154 🙞	155 🙜	156 🙝	157 🙟	158 ·	159 •
A0	160 ▪	161 ○	162 🞆	163 🞈	164 ◉	165 ◎	166 🔿	167 ▪	168 ◻	169 🟂	170 ✦	171 ★	172 ✶	173 ✴	174 ✹	175 ✵
B0	176 ⯐	177 ⌖	178 ⟡	179 ⌑	180 ⯑	181 ✪	182 ✰	183 🕐	184 🕑	185 🕒	186 🕓	187 🕔	188 🕕	189 🕖	190 🕗	191 🕘
C0	192 🕙	193 🕚	194 🕛	195 ⮰	196 ⮱	197 ⮲	198 ⮳	199 ⮴	200 ⮵	201 ⮶	202 ⮷	203 🙪	204 🙫	205 🙕	206 🙔	207 🙗
D0	208 🙖	209 🙐	210 🙑	211 🙒	212 🙓	213 ⌫	214 ⌦	215 ⮘	216 ⮚	217 ⮙	218 ⮛	219 ⮈	220 ⮊	221 ⮉	222 ⮋	223 🡨
E0	224 🡪	225 🡩	226 🡫	227 🡬	228 🡭	229 🡯	230 🡮	231 🡸	232 🡺	233 🡹	234 🡻	235 🡼	236 🡽	237 🡿	238 🡾	239 ⇦
F0	240 ⇨	241 ⇧	242 ⇩	243 ⬄	244 ⇳	245 ⬁	246 ⬀	247 ⬃	248 ⬂	249 🢬	250 🢭	251 🗶	252 ✔	253 🗷	254 🗹	255 ⊞

PARAMETER (OR 'PRESETS') FILE

To remove the nuisance of having to retype the same parameter settings every time you attach the indicator to a chart, you have two options. EITHER:

1. Use MT4 templates (but these overwrite any other indicators, objects, etc that you also have on the chart). In this case, the indicator settings are defined when you attach or edit the indicator, using MT4's built-in dialog; OR

2. Use parameter file(s), as follows:

- If you enter nothing or TXT into the **ParameterFile** setting, then the indicator looks for a parameter file called Presets---PSA.TXT in the ...../MQL4/Files folder.
- If you enter any other text (e.g. XXX, 001, etc) into the **ParameterFile** setting, then the indicator looks for a parameter file called Presets---PSA.XXX, Presets---PSA.001, etc in the ...../MQL4/Files folder. (This allows you to set up a different definition in each file, then simply enter the file suffix when you attach the indicator).
- If the file can't be found, then the settings in MT4's dialog will be used. If the file is found, then the settings in the file will be used instead. Hence if you want to force the use of the MT4 dialog settings, enter the name of a non-existent file (e.g. 'NONE').

Use a text editor like Notepad to view and/or edit the parameter presets files.

To disable an individual entry in the file, start the line with two slashes (//). This means that the setting specified in the MT4 dialog (when you attach/edit the indicator) will be used instead. To re-enable it, remove the slashes. If you have multiple 'unslashed' entries in the file, the last will be used. You can also disable an entry by setting its parameter value to an asterisk (*). Thus either of

```
// FinalScaleValue = 100;
```

or

```
FinalScaleValue = *;
```

will allow you to adjust the scale via the MT4 dialog, i.e. without the need to edit the parameter file.

You may use any number of embedded spaces or underscores, and/or any combo of upper or lower case, to aid readability, however you wish. The only exceptions are date fields (currently, there aren't any), which must be entered exactly in the form YYYY.MM.DD<space>HH:MM

Every line in the file (including blank lines, and lines beginning with two slashes) MUST end with a semi-colon (;). Failure to do this will cause unpredictable results.

If an entry occurs multiple times in the file, the last occurrence is used to set the parameter value, e.g.

```
FinalScaleValue = 50;
FinalScaleValue = 100;
```

means that the scaling will be set to 100.

To save typing and improve accuracy, copy/paste the sample lines in the file.

To disable the entire file, either rename it, or delete its content.

To have any changes take immediate effect, either (1) re-attach the indicator to your chart; (2) edit the indicator (press Ctrl-I); (3) switch between timeframes, or (4) switch between currency pairs. Otherwise the chart will not be refreshed/re-plotted until the **RefreshPeriod** setting takes effect.

Appendix 1 — index of additional features

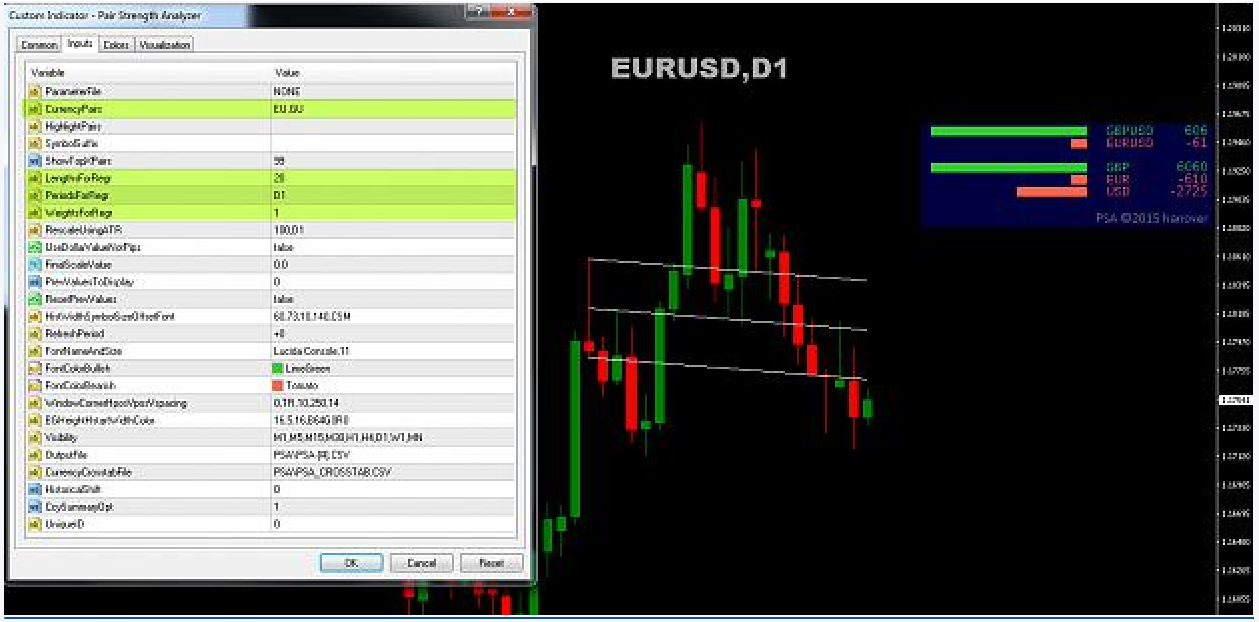
Detailed explanation of how the **LengthsForRegr**, **PeriodsForRegr** and **WeightsForRegr** settings work

OK let's start with the basics and then gradually work towards a more complex scenario.

We are looking at two currency pairs: EU and GU. *LengthsForRegr* is set to **20**, so we are looking at 20 period regression line only, and *PeriodsForRegr* is set to **D1**, so we are looking at the daily chart only. As can be seen, PSA returns a value of -61 for EU and +606 for GU. Now look at the 20 period regression lines for each of these two currency pairs in the first two screenshots. EU is sloping slightly downward (hence the negative value) while GU is sloping steeply upward (hence the positive value). The PSA values tell us that the slope of GU is approximately 10 times as steep as the slope for EU (606 = ~ 61 x 10).

Notes:
1. The values are not a measure of pips or time, simply a **relative measure** of the slopes between the currency pairs in question.
2. The PSA's summary values for the **currencies** are the average for each currency across all of the pairs, multiplied by 10 (to remove decimals and generate a nice round whole number). The value is inverted for the second named currency in the pair.

> *Attached Image (click to enlarge)*



Custom Indicator - Pair Strength Analyzer

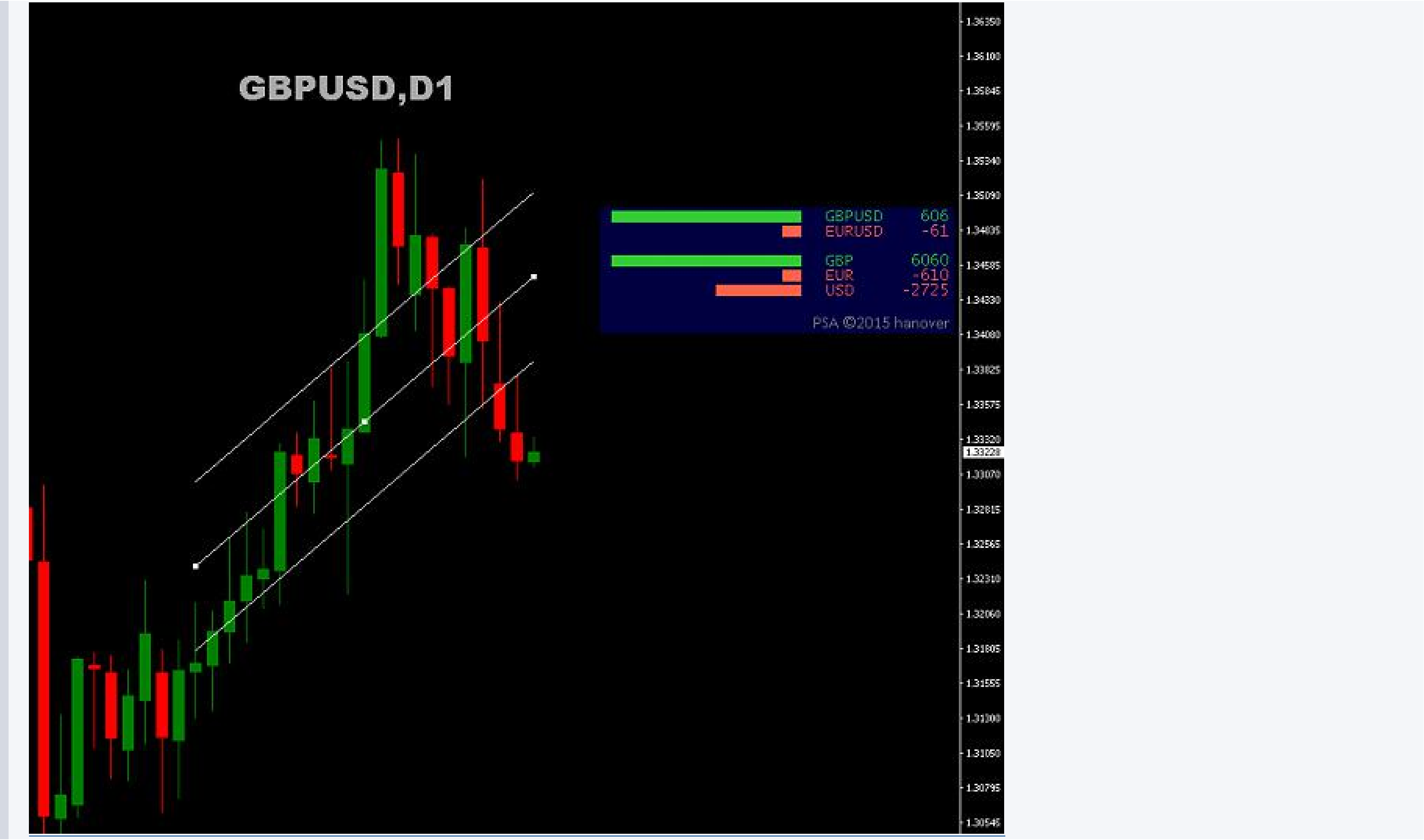
Variable	Value
ParameterFile	NONE
CurrencyPairs	EU,GU
HighlightPairs	
SymbolSuffix	
ShowTopNPairs	99
LengthsForRegr	20
PeriodsForRegr	D1
WeightsForRegr	1
RescaleUsingATR	100,D1
UseDollarValueNotPips	false
FinalScaleValue	0.0
PrevValuesToDisplay	0
ResetPrevValues	false
HistWidthSymbolSizeOffsetFont	60,73,18,140,CSM
RefreshPeriod	+0
FontNameAndSize	Lucida Console,11
FontColorBullish	🟩 LimeGreen
FontColorBearish	🟥 Tomato
WindowCornerHposVposVspacing	0,111,10,250,14
BGHeightHstartWidthColor	16,5,16,B64G0R0
Visibility	M1,M5,M15,M30,H1,H4,D1,W1,MN
OutputFile	PSA\PSA-[R].CSV
CurrencyCrosstabFile	PSA\PSA_CROSSTAB.CSV
HistoricalShift	0
CcySummaryOpt	1
UniqueID	0

EURUSD,D1

```
GBPUSD    606
EURUSD    -61

GBP      6060
EUR      -610
USD     -2725
PSA ©2015 hanover
```

Attached Image (click to enlarge)



GBPUSD,D1

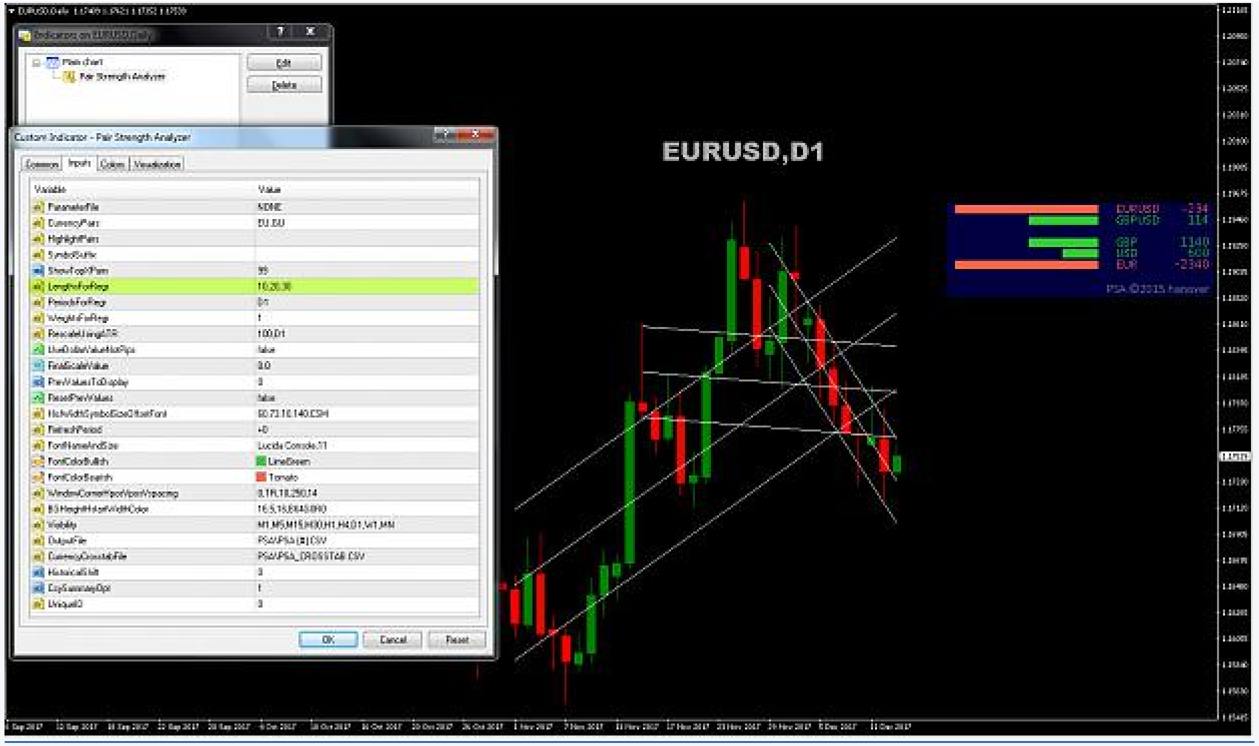
GBPUSD 606
EURUSD -61

GBP 6060
EUR -610
USD -2725

PSA ©2015 hanover

In the following screenshot I've changed *LengthsForRegr* to **10,20,30** which means that PSA now put three imaginary regression lines of lengths 10, 20 and 30, and calculates the average slope of the three lines for EU,D1, giving a value of -234 (downward. It's obvious from the screenshot that the downsloping 10 period line is steepest, affecting the average accordingly). I haven't shown the screenshot for GU,D1, but PSA shows that its average is +114 (upward).

> *Attached Image (click to enlarge)*



Custom Indicator - Pair Strength Analyzer

Common | Inputs | Colors | Visualization

Variable	Value
ParameterFile	NONE
CurrencyPairs	EU,GU
HighlightPairs	
SymbolSuffix	
ShowTopXPairs	99
LengthsForRegr	10,20,30
PeriodsForRegr	D1
WeightsForRegr	1
RescaleUsingATR	100,D1
UseDollarValueNotPips	false
FinalScaleValue	0.0
PrevValuesToDisplay	0
ResetPrevValues	false
HtAndWidthSymbolSizeOffsetFont	60,73,10,140,CSH
RefreshPeriod	+0
FontNameAndSize	Lucida Console,11
FontColorBullish	🟩 LimeGreen
FontColorBearish	🟥 Tomato
WindowCornerHposVposVspacing	0,TR,18,290,14
BGHeightHposWidthColor	16,5,18,8840,000
Visibility	M1,M5,M15,M30,H1,H4,D1,W1,MN
OutputFile	PSA\PSA_[#].CSV
CurrencyCrosstabFile	PSA\PSA_CROSSTAB.CSV
HistoricalShift	0
CsvSummaryOpt	1
UniqueID	0

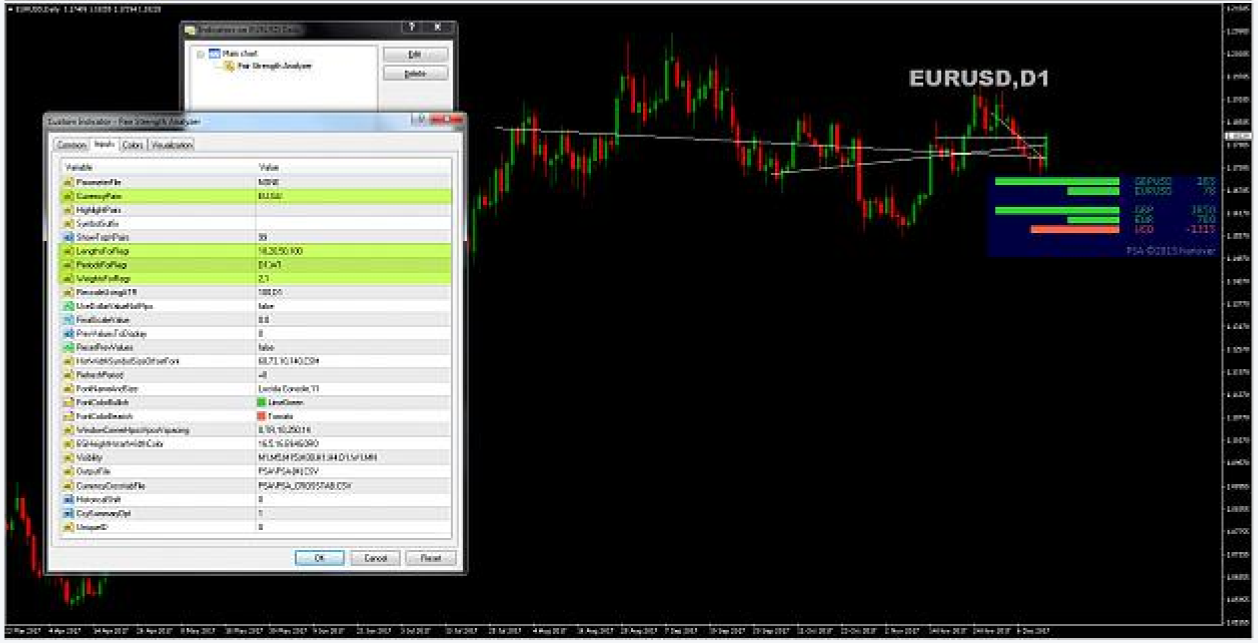
OK | Cancel | Reset

EURUSD	-234
GBPUSD	114
GBP	1140
USD	600
EUR	-2340

Now suppose we wanted to double the weight that PSA applies to the 10 period Regression line, to give more emphasis to the price movement of the most recent period. Then we would set *LengthsForRegr* to **10,10,20,30**. (Note that you can have up to 50 entries in *LengthsForRegr*, separated by commas. This determines the number of imaginary regression lines, and their lengths, for which PSA will average their slopes).

PSA is also capable of producing a consolidated value for regression lines over different timeframes. You simply enter the different time frames in *PeriodsForRegr*, separated by commas. For example, supposing *LengthsForRegr* is **10, 20, 50, 100** and *PeriodsForRegr* is **D1,W1**. Now PSA will average the slopes of 8 different regression lines, that is 10, 20, 50 and 100 period lines on the D1 timeframe, and likewise 10, 20, 50 and 100 period lines on the W1 timeframe, and print the consolidated average. This assumes that *WeightsForRegr* is set to **1,1**. If, for example, you wanted the daily component to have twice the weight of the weekly component, you would set *WeightsForRegr* to **2,1**. (Each entry in *WeightsForRegr* matches the corresponding entry in *PeriodsForRegr*). In the screenshot, you can see the settings, and also the 10, 20, 50 and 100 period regression lines for EU (on the daily chart only):

> *Attached Image (click to enlarge)*



Variable	Value
ParameterFile	NONE
CurrencyPairs	EU,GU
HighlightPairs	
SymbolSuffix	
ShowTopNPairs	99
LengthsForRegr	10,20,50,100
PeriodsForRegr	D1,W1
WeightsForRegr	2,1
RescaleUsingATR	100,D1
UseDollarValueNotPips	false
FinalScaleValue	0.0
PrevValuesToDisplay	0
ResetPrevValues	false
HorVertWidthSymbolSizeOffsetFont	60,71,10,140,234
RefreshPeriod	-8
FontNameAndSize	Lucida Console,11
FontColorBullish	🟩 LimeGreen
FontColorBearish	🟥 Tomato
WindowCornerHposVposVspacing	0,99,10,250,16
BGHeightHorzVertWidthColor	165,16,6946090
Visibility	M1,M5,M15,M30,H1,H4,D1,W1,MN
OutputFile	PSA\PSA-{#}.CSV
CurrencyCrosstabFile	PSA\PSA_CROSSTAB.CSV
HistoricalShift	0
CcySummaryOpt	1
UniqueID	0

The values of +185 for GU and +78 for EU tell us that GU has the higher average upward slope.

Please note that the timeframe entries in *PeriodsForRegr* can be either relative or absolute. This works the same as the *RefreshPeriod* as explained here.

The *WeightsForRegr* entries don't have to be double or triple, you can use decimal numbers, e.g. **1.5**.

Also, for more info, please see this post, and for additional examples my reply to Angela here.

The **RescaleUsingATR** setting

You can add any instruments/symbols (exotics, metals, commodities, indices, cryptocurrencies, etc) to the list in the *CurrencyPairs* parameter, provided that your broker has made them available. However, keep in mind that '1 pip' for a currency pair is not necessarily the same '1 pip' measure for another instrument, hence the slope values that PSA prints might not always represent an 'apples to apples' comparison. Moreover, even within the realm of currency pairs, different instruments can have significantly different average daily ranges. To help alleviate this problem, you can use the *RescaleUsingATR* setting (as explained in the instruction PDF attached to post #1) to have PSA scale the slope values relative to the ADR of the relevant instrument. But with the proviso that the values that you're comparing now reflect the extent to which the instrument is overperforming or underperforming its normal ADR, as opposed to its 'raw' slope value.

To further explain its use, the *RescaleUsingATR* setting requires two values, separated by a comma. If you leave the setting completely blank, then no re-scaling is performed, i.e. the 'raw' slope values are used. **This means that currency pairs with large daily ranges will tend to score higher.** However, if you enter the two values **nnn,timeframe** into *RescaleUsingATR*, then each slope value is **divided by** a nnn period ATR of the specified timeframe, arguably making for a more accurate comparison. The latter entry may be either an absolute or relative timeframe, as explained here. The values themselves aren't as important as the mere fact that any wide ADR variations are at least being mitigated, but FWIW as a guide I tend to use **100,D1**, i.e. a 100 period average daily range.

If you set *RescaleUsingATR* to **%,nnnn** (i.e. **percent sign**, comma, then a number) then instead of using regression line slopes, PSA will use the **percentage** gain (or loss) in the value of the instrument, from the close of each candle specified in *LengthsForRegr* to the current price. So for example, if *LengthsForRegr* is set to **10, 20**, then PSA will calculate the %gain from the close of the candle 10 periods ago to the current price, the %gain from the close of the candle 20 periods ago to the current price, average them, and then multiply the result by the **second value** that you entered into *RescaleUsingATR*. If, for example, the second value is 100, then the value displayed by PSA will be the actual %gain, e.g. a 2% gain would print as 2. Hence I recommend entering a value like **10000**, which would then show the gain in hundredths of a percent, e.g. 2% would print as 200.

The settings *PeriodsForRegr* and *WeightsForRegr* continue to operate as before, but using the %gain data rather than regression line slopes.

The goal of the new option is to allow improved 'apples to apples' comparison, i.e. %gain/loss, between forex and non-forex related instruments that use different units of measure, and daily ranges.

The **ResetPrevValues** setting

This is the way the indicator is intended to (and should now) work:

-- If *ResetPrevValues* is set to FALSE, and you manually refresh the chart (by changing either the instrument, or the chart timeframe; or by editing the parameter settings), then all of the values should move one column to the right, to allow a newly calculated value to be displayed in the leftmost column. This should also occur if the chart is refreshed automatically, as determined by the *RefreshPeriod* parameter.

-- If *ResetPrevValues* is set to TRUE, and you manually refresh the chart, then a reset is performed, causing the newly calculated value to display **in all of the columns**.

-- If you either close the chart, or remove the indicator from the chart, then everything likewise gets reset (which will be apparent only when the indicator is later reattached to a chart).

All of the above applies if *PrevValuesToDisplay* is set to a **positive** value (1 or 2). If it is set to a **negative** value (-1 or -2), then the above still applies, except that **differences** are displayed in the rightmost columns -- showing the increase or decrease since the last refresh(es) -- instead of the previous values.

Appendix 2 — Other updates

The **PrevValuesToDisplay** setting

Updated Jan 10, 2015 — indicator now contains these new features:

1. *PrevValuesToDisplay* setting: Previous values can now be displayed as differences (increase/decrease from prior value) instead of the actual value itself.

2. The optional Histogram settings are now much more customizable. Any Wingdings symbol may be used as the bar character (see chart of Wingdings symbol codes here).

3. Parameters can now optionally be entered via a parameter/presets file. See earlier "PARAMETER (OR 'PRESETS') FILE" section in this manual for more information.

The **HighlightPairs** and **ShowCcySummary** settings

Updated Dec 6, 2015 — indicator now contains these new features:

1. *HighlightPairs* setting: PSA will now place a user-defined highlight mark beside any pairs you choose, making it easier to find them as they move up/down the list.

2. *ShowCcySummary* has been replaced by *CcySummaryOpt*: there are now 4 different options for the Currency Summary Table.

Updated Sept 5, 2017 — indicator now contains these new features:

If you set *CcySummaryOpt* to:
0 = currency summary options (at foot of list) is not displayed
1 = each currency value is the **sum** of all of its constituent pair values
2 = currency values are further scaled into the interval (-n,n) where n is the *FinalScaleValue* (assuming n>0)
3 = currency values are further scaled into the interval (0,n) where n is the *FinalScaleValue* (assuming n>0)
Currencies are always sorted from the highest positive value (most bullish) to the highest negative value (most bearish)

The **BGHeightHstartWidthColor** and **CurrencyCrosstabFile** settings

Updated Sept 29, 2017 — the PSA indicator (latest version -- see download file below) now contains these new features:

1. The histogram facility has been updated to allow a user-defined font. Hence the CSM font (see this post for more info) can be applied to give the result displayed in first screenshot in post #215. If the font entry is omitted, the indicator uses the 'Wingdings' font as its default, like it did previously.

(The effect in the screenshot is also created by setting the second entry to 73, which is the ASCII code for uppercase 'I').

2. A background panel facility is now available via the new parameter *BGHeightHstartWidthColor*. Entries, separated by commas, are
i) height (per row),
ii) horizontal start position (pixel number) -- this should closely match your Hpos entry in *WindowCornerHposVposVspacing*,
iii) width of background box,
iv) color of background box (can be any valid MT4 token, e.g. Maroon, MidnightBlue, SaddleBrown, or RnnnGnnnBnnn where 0<=nnn<=255 for red, green, blue components).
To have no background panel displayed, leave the parameter blank, and the indicator will operate as it did previously.
See screenshots in post #215 for examples.

3. An output file of the PSA's historical values (for currencies only, not pairs/instruments) is now available via the new parameter *CurrencyCrosstabFile*. Enter the name of the file, which will be output to the ..../MQL4/Files folder (or your specified subfolder). Leave the parameter blank, and no file will be output. File format is as shown in sample below, and a new row is appended to the file every time the PSA values are updated (*RefreshPeriod*), or the indicator is manually refreshed (use a text editor like Notepad to manually remove any unwanted rows). First entry on each row is the MT4 date/time that the row was output. If you're using multiple instances of the PSA (whether attached to the same or different charts), you should specify a different file name for each instance, to avoid the same file from being overwritten by multiple sources, causing unpredictable results.

Inserted Code

```
2017/09/18 01:00:02,AUD, -7886,CAD, -9771,CHF, -7199,EUR, -5234,GBP, 53589,JPY, -23520,NZD, 3160,USD, -72
2017/09/18 01:15:00,AUD, -8116,CAD, -9966,CHF, -6531,EUR, -4707,GBP, 53054,JPY, -23309,NZD, 3210,USD, 202
2017/09/18 01:30:02,AUD, -7854,CAD, -9916,CHF, -6451,EUR, -4574,GBP, 52946,JPY, -23479,NZD, 3340,USD, -1
2017/09/18 01:45:01,AUD, -8463,CAD, -9871,CHF, -6654,EUR, -4753,GBP, 53120,JPY, -23029,NZD, 3457,USD, 62
2017/09/18 02:00:01,AUD, -8421,CAD, -9770,CHF, -6630,EUR, -5000,GBP, 52863,JPY, -23007,NZD, 3639,USD, 189
2017/09/18 02:15:18,AUD, -8070,CAD, -9830,CHF, -6603,EUR, -4833,GBP, 52860,JPY, -23471,NZD, 3609,USD, 173
2017/09/18 02:30:02,AUD, -7764,CAD, -9947,CHF, -6581,EUR, -4851,GBP, 52584,JPY, -23371,NZD, 3564,USD, 204
2017/09/18 02:45:00,AUD, -7997,CAD, -10167,CHF, -6731,EUR, -5003,GBP, 52853,JPY, -23826,NZD, 3697,USD, 922
2017/09/18 03:00:00,AUD, -8127,CAD, -10176,CHF, -6819,EUR, -4696,GBP, 53071,JPY, -23764,NZD, 3387,USD, 926
2017/09/18 03:15:01,AUD, -7773,CAD, -10340,CHF, -6931,EUR, -4797,GBP, 53120,JPY, -23774,NZD, 3747,USD, 571
2017/09/18 03:30:04,AUD, -7727,CAD, -10294,CHF, -6419,EUR, -4596,GBP, 52941,JPY, -23879,NZD, 3420,USD, 356
2017/09/18 03:45:01,AUD, -7660,CAD, -10160,CHF, -7201,EUR, -4454,GBP, 53194,JPY, -24231,NZD, 3471,USD, 780
2017/09/18 04:00:01,AUD, -7336,CAD, -10246,CHF, -7049,EUR, -4330,GBP, 52979,JPY, -24529,NZD, 3726,USD, 504
2017/09/18 04:15:00,AUD, -7000,CAD, -10024,CHF, -7311,EUR, -4490,GBP, 52879,JPY, -24561,NZD, 3994,USD, 356
2017/09/18 04:30:01,AUD, -6244,CAD, -9823,CHF, -7251,EUR, -4619,GBP, 52787,JPY, -25073,NZD, 3984,USD, 12
```

An indicator like the 'Plot PSA Crosstab.ex4' (see download file below) can then be used to plot historical values that were previously displayed by the PSA (see screenshot in post #215). Note that if the PSA's *RefreshPeriod* is different to the chart TF, then the values plotted won't align under the correct candles on the chart.

The **HighlightPairs** and **SmoothnessMultiplier** settings

Updated Aug 26, 2018 — I have made two enhancements (download the new version in the link below):
1. If you enter an asterisk (*) as the second parameter of *HighlightPairs*, the current chart's pair will be highlighted in the list displayed by PSA.
2. New parameter setting: *SmoothnessMultiplier*.
-- If set to 0, PSA operates as it did before.
-- If set to a positive (>0) value, PSA will calculate a standard deviation of all of the regression line slopes, in an attempt to establish which is the **smoothest** (as opposed to steepest) trending pair. Each of these values is then multiplied by your *SmoothnessMultiplier* value, before being displayed as an **additional column** at the right of the list (sorry, there is currently no facility to sort the list by the values in this column). If you use *RescaleUsingATR*, the slope values are likewise rescaled before their standard deviation is calculated. The lower the standard deviation value displayed, the smaller the slope deviation between the lines (the closer the displayed value gets to 0, the closer all of the lines are to being exactly the same slope) and hence the smoother the price movement. And *vice versa*.

More on the **LengthsForRegr**, **PeriodsForRegr** and **WeightsForRegr** settings

Updated Aug 5, 2019 — I've made PSA a little more bulletproof, to correct any user entries that are less than meaningful:
(1) Any missing, zero or negative values in the list of entries in *LengthsForRegr* are assumed to be 1. Also, if you leave *LengthsForRegr* completely blank, it's assumed to be a list with just one value: 1
(2) If you forget to match the number of entries in *PeriodsForRegr* and *WeightsForRegr*, any missing (or zero, or negative) values in *WeightsForRegr* are assumed to be 1
Everything else — the calculations, display, outputs, etc — remain unchanged.

Alerts facility

Updated Sept 7, 2019 — New alerts feature and settings added:

PSA is now capable of generating both chart (audio/visual) and email-based alerts. Push notifications are not available, because the indicator was written prior to build 600. I'm not intending to provide b600 or MT5 versions. The email based alerts assume that you've completed the Email tab in the MT4 Options settings correctly (contact your telecoms provider if you need help with SMTPs etc).

There are 4 new parameter settings:

```
bool ChartAlerts = false;
```

Controls whether or not audio/visual 'popup' alerts will occur.

```
string AlertEmailSubject = "";
```

If blank, no email alerts will occur. If non-blank, this text will appear in the email subject, and the alert message in the email body.

```
int AlertThreshold = 0;
```

This determines at what value alerts will trigger, for items in the pairs/instruments section. For example, if set to 90, an alert will be generated whenever the printed value has moved from < 90 on the previous refresh, to >= 90 on the current refresh; or the printed value has moved from > -90 on the previous refresh, to <= -90 on the current refresh. The refresh frequency is controlled by the PSA's *RefreshPeriod* setting.
If alerts for multiple instruments occur simultaneously, they are all gathered up and posted as a single alert.
To stop alerts from being triggered for pairs/instruments, leave this set at the default value of 0.

```
int AlertThresholdCcy = 0;
```

This works exactly the same way as *AlertThreshold*, except that it operates for the currencies section.

To confirm that it's working correctly, set *PrevValuesToDisplay* to 1. Then you can see what the values were on the previous refresh, and compare them to the current one. Note that if a value fluctuates above and below a threshold value repeatedly between refreshes, repeated alerts will be generated.

Everything else should work as it did before. I've tested the alerts as best as I can, but it's the weekend now and markets are closed. I'm away on vacation until the end of the month.

Enhancement to **FinalScaleValue** setting

I have just made a small enhancement; here is how *FinalScaleValue* works now:

-- If its value is 0 or 1, the calculated values are displayed as is, without any scaling.

-- If its value is >1, then the calculated values are scaled to fit the value you input. For example, if you input 1000, then the highest (absolute) value is scaled up or down to +/- 1000, and all other values are scaled by the same factor.

The sign of the value you input (+ or –) determines the way the displayed data is sorted:

-- If this value is **positive**, the data is sorted so that the highest positive/negative data (strongest uptrend/downtrend) displays at the top of the list, and the values taper down toward 0 (price is moving sideways) at the bottom of the list.

-- If this value is **negative**, the data is sorted so that the highest positive data (strongest uptrend) displays at the top of the list, and the highest negative data (strongest downtrend) at the bottom of the list.

This applies only to the **pairs** display.

The **currencies** will always display from the highest positive data (strongest currency) at the top of the list, to the highest negative data (weakest currency) at the bottom of the list.

The **PlotRegressionLines** and **LineColorForTF** settings

Updated March 3, 2020 — PSA now includes the option of actually plotting the regression lines, from which its calculations are based, on the price chart. This allows you to see the slopes and weights of the lines that contribute to the final printed value. The new parameter settings are:

PlotRegressionLines -- if set to true, the lines will be plotted.

LineColorForTF -- enter up to 9 colors, separated by commas, that correspond the timeframes you've specified in *PeriodsForRegr*. The lines are always solid, and their width is determined by your *WeightsForRegr* entries. You can enter valid MT4 tokens (e.g. Crimson, Goldenrod, SaddleBrown, MediumSeaGreen, DarkSlateGray, etc -- see the color chart here) or **RnnnGnnnBnnn** where each **nnn** is a value between 0 and 255 representing the red, green and blue components of the color (e.g.B255G255 is Aqua).

Example: if
PeriodsForRegr = **H4,D1**
WeightsforRegr = **5,1**
LineColorForTF = **White,Yellow**
Then the regression lines plotted for H4 will be white, width 5; while the regression lines for D1 will be yellow, width 1. Lines are plotted for each entry in *LengthsFor Regr*, and are shifted up/down (without changing their slope) so that they all converge at the current price. Of course this applies to the current chart's pair only. However, you can change to charts of different pairs, and compare all of the slopes that are averaged to give the final consolidated value.

If you set *RescaleUsingATR* to %, then PSA simply plots the trendlines from the closing price of the historical candle, to the current price, reflecting the % change (note that for timeframes other than the current chart, the lines are adjusted so that they plot from where the closing price of the candle on the selected TF (in *PeriodsForRegr*).)

If you set *PlotRegressionLines* back to false, the lines will vanish from the chart.

The **ShowAngleText** and **ShiftLinesToConvergence** settings

Updated March 6, 2020 — Fixed a couple of small bugs. Also added new parameters:

ShowAngleText -- if true, displays the angle (or movement %, if *RescaleUsingATR* = **%**) above each line, if you have MT4's 'Show object descriptions' (type F8 for Properties/Common tab) checked ON; or in the help popup bubble when you hover the mouse cursor over a line, if checked OFF.

ShiftLinesToConvergence -- if true, and *RescaleUsingATR* is **not** = **%**, the lines are shifted up/down (without changing their slope) so that they all converge to the current price. If false, the regression lines are not shifted. Has no effect for *RescaleUsingATR* = **%,** since the lines converge to the current price regardless**.**

The **SmoothSlopeUsingLogarithms** setting

Updated March 14, 2020 — Added new parameter setting *SmoothSlopeUsingLogarithms*. Applies only if *RescaleUsingATR* is **not** = **%**. Reduces the calculated slope by appying a natural logarithm to its value, thereby reducing the variance between different slope values.

Appendix 3 — limitations of price-based strength meters

Which is the best currency strength meter, and why?

I just discovered this site. It is one of many that either sells or promotes paid-for currency strength meters (CSMs). A little commonsense should tell you that it's decidedly suspect.

They promote four different paid-for CSMs, and claim that none of the free products that they tested were up to standard. On what grounds? They state that accuracy is important. Of course it is. But, assuming that there are no coding errors in the indicator calculations, accuracy is dependent solely on the price data being provided by the br0ker or market maker.

Which brings me to my next point(s). Firstly, all CSMs are derived from price, and price lags sentiment (more here). In an industry where getting quickest access to quality information can provide a competitive advantage, institutional traders tend to use indicators that quantify sentiment, economic flows, orderflow/liquidity etc. (Many of these indicators are proprietary and lie beyond the scope of 'retail' charting packages like MT4).

Secondly, all price analysis is retrospective, and as price moves as waves within waves, this opens the door for arbitrariness that can only be resolved by selective bias. To explain with an example: the last 10 candles might represent an up-move, but these may be the most recent part of a larger sideways move, which in turn may be part of a larger down-move. Hence, like all other technical measures (e.g. moving averages, regression lines, etc), the price-derived CSM measure is determined arbitrarily by the historical horizon that's under study. There is no objective way of proving that one horizon is superior to another, that one way of calculating the value is superior to another, and hence that one CSM is necessarily superior in its 'accuracy' to another. All are ultimately giving a retrospective view of what price has done, and have no greater predictive value than price itself (or any other derivative of price).

Thirdly, predictive value comes down to the strength of the underlying sentiment which is determined by a number of fundamental reasons, many of which are economic, structural and/or logistical. If, for example, a CSM says that EUR is weak and USD strong, then the downtrend in EURUSD will continue only if there are overriding reasons -- i.e. reasons that underlie price -- causing it to do so. A price based CSM is not equipped to take these fundamental factors into account.

Fourthly, a CSM is merely a tool, and it is down to the trader, and how s/he interprets the strength/weakness measure, and uses it in the context of an overall trading system, that is an ultimate determinant in success or failure.

Another claim made by the website I targeted is that their recommended CSMs have more features than the free offerings. Just for starters, compare them to the PSA, Recent Strength and CSM indicators that I wrote, and you'll see that this claim is completely false. I have no reason to endorse my own creations, they are offered FREE, and I have no personal interest in who does or doesn't use them. And there are some equally good FREE CSMs developed by other programmers, some of whom are regular contributors to the FF forums (see the links in the green typeface here, for example).

Hence my point is this: Why spend money on CSMs, when
(1) all CSMs are price derived, for which there are inherent limitations;
(2) no one strength calculation is necessarily superior, in terms of its predictive value, to another; and
(3) there are plenty of FREE products that are every bit as accurate, and even more feature rich, than the expensive CSMs being recommended in the website.

====== END OF DOCUMENT ======