

# Recent SR indicator

## Overview

This is the most sophisticated support/resistance (S/R) indicator that I have created. It highlights some of the difficulties in translating what appears obvious to the technical analyst's naked eye, into a mathematical formula that can be coded. For better or worse, automating the S/R process introduces a level of objectivity: the indicator may pick up S/R zones that somehow escaped the analyst's view.

The indicator automatically plots horizontal S/R bands on your MT4 charts, according to the parameter settings that you enter. It does NOT plot any of diagonal trendlines, Fibon retracement levels, pivot levels or round numbers. Nonetheless, the indicator is highly configurable, necessitating a learning curve. Timeframe, bandwidth, frequency levels, visibility level & clearance can all be customized, resulting in very different S/R plots. Experiment with different parameter settings until you find the generated S/R bands approximating those that feel 'right' for you.

Each S/R band is actually an MT4 object (rectangle). You can have the indicator automatically maintain these on a going-forward basis, or simply have them plotted once, after which you can move or delete them, and/or add your own S/R. When you delete the indicator from the chart, all of the automatically generated bands are deleted also.

It is possible to have the bands generated according to candle data in one timeframe (TF), and then have these plotted on a different TF. Thus you could have H4 S/R levels being plotted, and 'locked in place', while you switch to different TFs (e.g. M5, M15, M30, etc).

The bands may be plotted 'behind' the price candles in the 'body' of the chart, and/or as a frequency 'histogram' (showing how many times price reached each S/R band) in the empty space at the right of the chart. You can also optionally have the indicator plot a 'Market Profile' type histogram, showing how many times (candles) price traversed through each level. It can also be used to plot 'fractals' (local higher highs and lower lows). You can use different colors (or shades of a color) to highlight different frequencies.

If you want to have different 'types' of S/R represented on the same chart, then simply attach multiple instances of the indicator. For example, you could have high visibility, wider bandwidth, longer TF S/R plot in shades of one color, and lower visibility, narrow bandwidth, short TF S/R plot in shades of another.

The indicator 're-paints' in the sense that as new S/R points are encountered, these (depending on your parameter settings) may not be created until all of your stipulated criteria have been met. Thus a currently displayed band that appears to have originated in the past might not have been displayed at that historical point. Since this indicator is unlikely to be used for generating entry signals, this should not be an issue.

Parameter settings may be entered either through MT4's dialog window, when you attach or edit the indicator, or via parameter file(s).

An output file may also optionally be generated, detailing the frequency of qualifying points at each S/R level. This file could then be used by an EA or imported to an application like Excel for further analysis.

I will possibly add an 'alerts' facility (when price reaches or approaches an S/R band) in a subsequent release.



## Conceptual Outline

Imagine a grid being overlaid on your MT4 chart (see above). The width of each 'cell' is one candle. The height of each cell in the grid is the *BandwidthPips* setting that you specify (thus you are 'slicing' the chart horizontally into bands of *BandwidthPips* pips each). Then the number of qualifying points (maximum of one 'QP' per cell) is summed horizontally for each row across the grid. The color of the box plotted is determined by the *FrequencyLevel* and *FrequencyColor* settings. For example if *FrequencyLevel* = "1,5,10" and *FrequencyColor* = "Maroon,Crimson,OrangeRed" then rows that have 1 or more, 5 or more, or 10 or more QPs will be over-plotted with zones (MT4 rectangles) of progressively brighter shades of red. See notes on color palette below.

A **qualifying point** (QP) may be one or more of:

- A candle high – if the *CandleHighsValid* setting = TRUE
- A candle low – if the *CandleLowsValid* setting = TRUE

- A candle close – if the *CandleClosesValid* setting = TRUE
- Any point at all on the candle – if the *AllCandlePointsValid* setting = TRUE (this is useful for generating a 'Market Profile' type plot).

Depending on these settings, if a candle qualifies on more than one point, only one QP may be 'scored' per **cell**.

The candle highs and lows optionally have additional filtering criteria:

- *VisibilityLevel* – if set to **0**, all candles qualify. If set to **1**, a candle high must be flanked by at least **one** immediately lower high on each side, in order to qualify. If set to **2**, a candle must be flanked by at least **two** immediately lower highs on each side. And so on. Vice versa for candle lows. (This is the same concept as Tom DeMark supply/demand points). If set to (for example) **2,1**, then a candle high must be immediately preceded by at least 2 lower highs, and immediately followed by at least one lower high. And so on.
- *ClearancePips* – this will filter out even more candle highs/lows. The candle high must exceed the immediately preceding and following candles (determined by *VisibilityLevel*) by at least *ClearancePips* pips, to qualify. A high *ClearancePips* value will ensure that only 'spikes' will qualify.

## Dataset parameters

*TimeFrame* – determines which TF's candles will be used. Leaving this setting blank defaults to the current chart's TF (thus when you switch TFs, a new set of S/R zones will be plotted). Typing one of M1, M5, M15, M30, H1, H4, D1, W1 or MN (upper or lowercase doesn't matter) will ensure that this S/R for the specified TF will remain locked on the chart, even if you switch TFs. This makes it possible to view (for example) key D1 S/R levels on an intraday chart. If you want to overlay multiple TF S/R levels, attach multiple occurrences of the indicator (see 'Multiple instances' section below).

*LookbackCandles*, *StartDateTime* – these determine the data window ('DW'), i.e. how many candles, backward from the currently forming candle, that you wish to have processed. (Obviously the greater the number of candles, the slower your computer will perform).

A positive (>0) setting in *LookbackCandles* means that the last *LookbackCandles* candles will be processed, creating a 'rolling' DW that drops off a leftmost candle, every time a new candle (on the S/R TF) forms. In this case, the *StartDateTime* setting is ignored.

A negative (<0) setting in *LookbackCandles* means that the *StartDateTime* setting will be used instead. The first (leftmost) candle in the DW will remain fixed at the date/time entered (required format is **YYYY.MM.DD**<space>**HH:MM**), and as new candles form, the DW will increase in size.

*IncludeCurrentCandle* – TRUE means that the currently forming candle (in the S/R TF) will be included in the DW. FALSE means that the DW ends with the penultimate candle (second from the right) in the chart.

*RefreshEveryXMinutes* – a negative value (<0) means that no auto-refresh will take place, i.e. the auto-created S/R bands will be plotted only when the indicator is first attached. This allows you to then move or delete auto-created S/R bands, and/or add your own. A **0** value means that an auto-refresh will occur every tick (which could slow computer performance down, especially if the indicator has been attached to many charts). A value of 1, 5, 15, 30, 60 or 240 means that the indicator will be refreshed and re-plotted every time a new tick occurs on the M1, M5, M15, M30, H1 or H4 chart of this pair, respectively. Note that you can manually refresh by doing any of (1) switching between chart TFs; (2) editing the indicator settings (press Ctrl-I); or (3) deleting and re-attaching the indicator.

## Frequency and Color palette

*Frequency Levels* – these allow up to 9 different colors to be assigned, depending on how many QPs are met across each row of the grid. Numbers must be entered in **ascending** sequence, and be separated by commas, e.g. 1,5,10 will allow different colored bands to be displayed if 1 to 4 QPs exist, 5 to 9 QPs exist, or 10 or more QPs exist, on each row (band).

*FrequencyColors* – Color descriptions should likewise be separated by commas, matching the corresponding *FrequencyLevel* entry. Hence there should be the same number of entries for each of these parameters. Color descriptions may be any mix of upper or lowercase, and may include embedded spaces, to aid readability, if desired. Each entry may be set up in either one of two ways:

1. By typing a valid MT4 color descriptor (e.g. DarkGoldenrod, Lime, Burlywood, Silver, etc). To help you, the dummy *ColorChart* parameter may be used to display MT4's default palette (mousing over any tile in the chart displays its descriptor name); OR
2. By using a RGB descriptor, e.g. **R125G95B45** means a mix of Red=125, Green=95, Blue=45. Each value must be between 0 and 255 (inclusive). The R, G and B characters can be upper or lowercase, in any order, and may be left out (which results a zero value for that component). Hence **B60G95** means a mix of Blue=60, Green=95, Red=0. To help you, the dummy *ColorChart* parameter may be used. Click on the 'Custom' button, drag the mouse around until you have the desired color, and then note the Red, Green and Blue values at the lower right of the window. So if the desired color comprises Red=120, Green=115, Blue=84, then type **R120G115B84** (or **r120 g115 b84**) as the *FrequencyColors* entry.

An RGB descriptor allows you to easily define different shades of a 'pure' color. For example, a *FrequencyColors* entry of **R65, R105, R145, R185** will simply cause bands colored with increasingly brighter shades of red to be plotted, at the different *FrequencyLevels* that you specify.

## Output parameters

A separate box (S/R zone, or MT4 rectangle) is plotted for each band on the MT4 chart. This starts just to the left of the first QP in that band, and proceeds through to the currently forming candle. Box height is the height of the band (*BandwidthPips*). A histogram, showing the number of QPs in each band, can optionally also be plotted, to the right of the price candles (see below). This is useful for Market Profile type plots.

*FilledBoxes* – TRUE means that the S/R zones plotted will be filled. FALSE means that only the box perimeters will be plotted.

*HistogramSize* – if 0 (the default), no histogram will be plotted. If a positive number, then the next *HistogramSize* price points (on the current chart's TF) will be used to plot the histogram, and values will be scaled to fit, as best as MT4 granularity permits.

*HistogramFontSize* – if 0, no text annotation (number of QPs) will be plotted near the center of the histogram bar. If a positive number, then that determines the font size that will be used. *HistogramFont* and *HistogramFontColor* determine the font (Arial etc) and its color, respectively.

*HistogramPriceMask* – leave blank (the default), and the price at each level will not be shown. Enter a value such as 2.4 to have price displayed, formatted with 2 places to the left of the decimal point, and 4 to the right.

*ShowHistogramOnly* – TRUE means that the S/R bands will not be plotted across the chart prices (and possibly obscuring other objects on the chart). FALSE (the default) means that the S/R bands will be plotted 'behind' the price candles.

*OutputFile* – if left blank (the default), no output file will be generated. A non-blank entry will have a file of the same name created (in the `....\experts\files` folder), and then rewritten in its entirety every time an auto-refresh occurs (see *RefreshEveryXMinutes* parameter). The file is written in CSV format, showing the pair, S/R TF, price band levels, and frequency (number of QPs) at each band level. This file could then be imported into an EA, or a product like Excel, for further analysis.

## Parameter file(s)

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:

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.
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---Recent SR.TXT** in the ...\**experts\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---Recent SR.XXX**, **Presets---Recent SR.001**, etc in the ...\**experts\files** folder. (This allows you to set up a different S/R 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, simply enter the name of a non-existent file.

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

To disable an entry in the file, start the line with two slashes (/). (This means that the setting specified in the MT4 dialog will be used.) To re-enable it, remove the slashes. If you have multiple 'unslashed' entries in the file, the last will be used.

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 exception is the StartDateTime, which must be entered exactly in the form **YYYY.MM.DD**<space>**HH:MM**.

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

To save typing and improve accuracy, copy/paste the sample lines in the file. To disable the entire file, 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); or (3) toggle between timeframes. Otherwise the chart will not be refreshed/re-plotted until the *RefreshEveryXMinutes* setting takes effect.

### **New parameters in v 1.01**

*IncludeNextBandAbove* – setting this to FALSE means that the indicator will behave as in previous versions. Setting it to TRUE is designed to work around the problem that I'm about to describe. As already explained, the chart is divided into horizontal sections of *BandwidthPips* pips each. So, for example, if *BandwidthPips* = 5, then section boundaries might be 0.9000, 0.9005, 0.9010, etc. Let's suppose we have QPs at 0.9004 and 0.9007. These QPs are within 5 pips of each other, but due to the arbitrary section boundaries, exist in different bands and therefore ordinarily wouldn't both accumulate in the same band frequency. However, by setting *IncludeNextBandAbove* to TRUE, all QPs between 0.9000 and 0.9010 will now be accumulated, frequency-wise, in the 0.9000 to 0.9005 band; all QPs between 0.9005 and 0.9015 will now be accumulated, frequency-wise, in the 0.9005 to 0.9010 band; and so on. Obviously this parameter is significant only if one is concerned with frequency type plots.

*PlotProximityPips* – setting this to a negative value means that the indicator will behave as in previous versions, i.e. all S/R bands falling within the limits of the price window will be visible. However, to conserve computer resources, this parameter can be set to a number of pips, and only S/R bands falling within the number of pips of the current bid price will be plotted. Thus S/R bands will appear and disappear from the chart as price moves, i.e. all bands will be recalculated and replotted as often as the *RefreshEveryXMinutes* parameter dictates.

*HistoricalShift* – this is, by default 0, which means that the indicator will behave as in previous versions. However, setting a value  $> 0$  will cause the plot to display as it would have *HistoricalShift* candles ago, given the specified *TimeFrame*.

### **Multiple instances of 'Recent SR' indicator on the same chart**

If you want to have different 'types' of S/R represented on the same chart, then simply attach multiple instances of the indicator. For example, you could have high visibility, wider bandwidth, longer TF S/R plot in shades of one color, and lower visibility, narrow bandwidth, short TF S/R plot in shades of another.

Use of different colors is important, and not just to visually distinguish the different S/R types. The potential problem exists that a price level might meet criteria in both instances of the indicator, causing **overlapping** bands to be plotted. The way in which MT4 'sums' colors might surprise you, for example red overlapping red creates black, which would 'disappear' in a chart with a black background.

Given that you want to overlay 3 different sets of colors, I choose the primary colors red, blue and green. So for the first instance of the indicator, I would set *FrequencyColors* to different shades of 'pure' red; the second instance, 'pure' blue; the third instance, 'pure' green. So when the bands overlap, we get:

- blue + green = cyan
- blue + red = magenta
- green + red = yellow
- blue + red + green = gray

All of these will display well enough on a black background chart.

Avoid these combinations:

- blue + blue = black
- red + red = black
- green + green = black

See the 'Frequency and Color palette' section above for information on how to enter the colors.

==== END OF DOCUMENT ====