

DayBreakTrail (v1.013)

MetaTrader 4 Expert Advisor

Copyright © 2014, Fernando M. I. Carreiro

Table of Contents

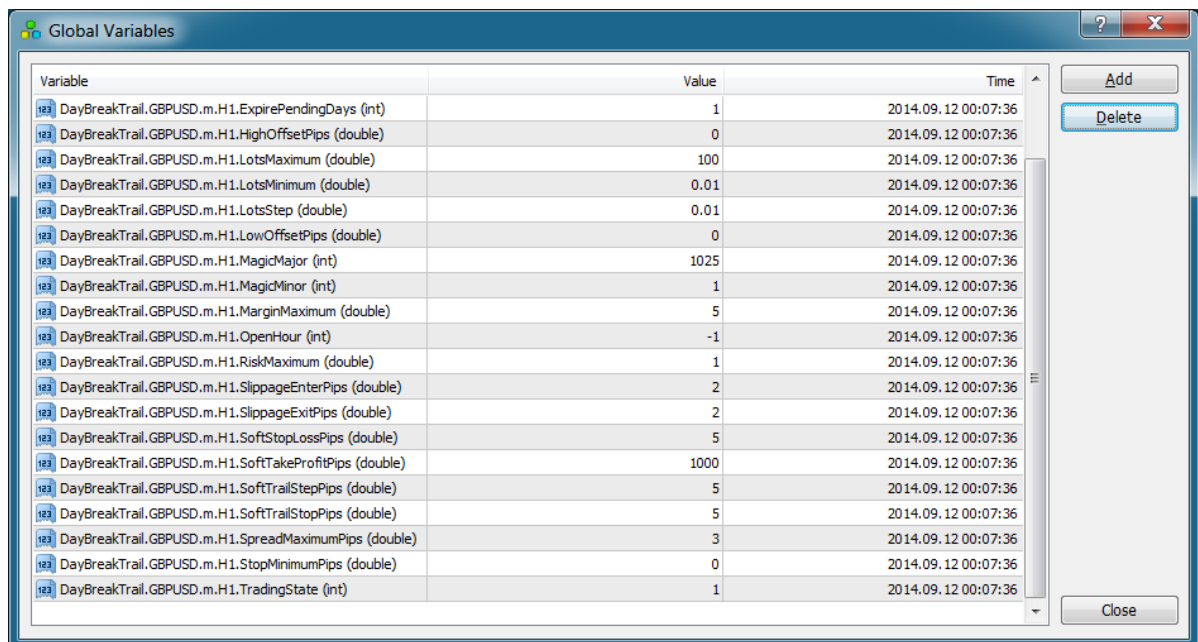
Inputs and External Variables	2
Parameter Settings.....	3
Global Variables	3
intExpirePendingDays	3
intExpireOpenHours	3
intOpenHour	3
dblHighOffsetPips, dblLowOffsetPips	3
dblSpreadMaximumPips	4
dblStopMinimumPips	4
dblSoftTakeProfitPips, dblSoftStopLossPips	4
dblSoftTrailStopPips, dblSoftTrailStopPips	4
dblLotsMinimum, dblLotsMaximum, dblLotsStep	5
dblSlippageEnterPips, dblSlippageExitPips	5
dblRiskMaximum, dblMarginMaximum	5
dblBalanceMinimum	6
intMagicMajor, intMagicMinor	6
Local Variables	6
intParamUpdateMins	6
intRetryCount, dblSleepTimeMean, dblSleepTimeMax	6

Inputs and External Variables

Most of the parameters (*inputs* and *external variables*), are stored as *MetaTrader Global Variables* after the *Expert Advisor* is first instantiated upon a *Chart* for particular *Symbol* and *Timeframe*. Only a few of the parameters are local and are set or modified by directly accessing the properties window for the *Expert Advisor* on the *Chart* in question.

After its first instantiation, the stored variables have to be changed or updated via the *Global Variables* management window, accessible by the menu option **Tools** → **Global Variables** or by hitting the **F3** key. *Global Variables* can remain stored for up to 4 weeks if not actively used or accessed, otherwise they will remain active even after a catastrophic PC reboot, allowing the *Expert Advisor* to pick-up and continue with the same settings, when it is reattached to a *Chart* of the same *Symbol* and *Timeframe*.

The *Expert Advisor* will read the stored *Global Variables* and refresh the internal values, on every new bar tick or on every timed interval set by the "*intParamUpdateMins*" parameter, described in more detail further on in this document.



Variable	Value	Time
DayBreakTrail.GBPUSD.m.H1.ExpirePendingDays (int)	1	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.HighOffsetPips (double)	0	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.LotsMaximum (double)	100	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.LotsMinimum (double)	0.01	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.LotsStep (double)	0.01	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.LowOffsetPips (double)	0	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.MagicMajor (int)	1025	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.MagicMinor (int)	1	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.MarginMaximum (double)	5	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.OpenHour (int)	-1	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.RiskMaximum (double)	1	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.SlippageEnterPips (double)	2	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.SlippageExitPips (double)	2	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.SoftStopLossPips (double)	5	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.SoftTakeProfitPips (double)	1000	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.SoftTrailStepPips (double)	5	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.SoftTrailStopPips (double)	5	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.SpreadMaximumPips (double)	3	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.StopMinimumPips (double)	0	2014.09.12 00:07:36
DayBreakTrail.GBPUSD.m.H1.TradingState (int)	1	2014.09.12 00:07:36

Parameter Settings

Global Variables

intExpirePendingDays

Although this *Expert Advisor* uses *Market Orders* instead of *Pending Orders*, it still uses the concept of an **Expiration Time** of equivalent *Pending Orders* set at the beginning of the opening session.

Set this parameter to the number of **days** in which to **expire** the *Virtual Pending Orders*.

intExpireOpenHours

Once a *Market Order* is opened, the order will expire or close after a set amount of time based on this parameter.

Set this parameter to the number of **hours** before which open orders will be **closed**.

intOpenHour

In order to use the *High* and *Low Breakout Prices*, based on the *New York* session close, the opening hour has to coincide with the **Opening of the Sydney Session**.

If set to do so, the *Expert Advisor* can attempt to calculate the opening hour **automatically** by looking up the opening hour of Weekly bars. This should in theory automatically compensate for *Daylight Savings* as well.

However, should the automatic function not evaluate the situation properly or if you prefer to set this manually, the easiest way to set this, if you don't know how to calculate *TimeZone* offsets or *Daylight Savings*, is just to monitor your broker's opening hour during the week opening, on Sunday evening or Monday morning (example: if your broker weekly session starts at 22h00 on Sunday evening, then the opening hour to use for this parameter is 22).

Set this parameter to the **hour** (0-23) of the *Sydney Session* opening. For automatic adjustment, use a setting of **-1** (negative one).

dblHighOffsetPips, dblLowOffsetPips

The *Breakout Prices* for the *Highs* and *Lows* can be offset by a certain number of *pips* as a way to create a certain buffer zone for the breakout

Set these parameters to the amount of *pips* you want to offset the *High* and *Low Breakout Prices*. Both positive and negative values are allowed for these parameters.

Please note, that since *Market Orders* are used, there is no need to compensate for the *Average Spread* on the *High Offset* value.

dblSpreadMaximumPips

Before a *Market Order* is placed, the current spread is checked to see if it is below the specified maximum allowable value defined by this parameter.

Set this parameter to the maximum allowable spread value in pips.

dblStopMinimumPips

The stop sizes for the *Take-Profit*, *Stop-Loss* and *Trailing-Stop*, are checked against this parameter to make sure they never go below a minimum value. If this parameter is below the current *Market Stop Level*, it is internally updated to respect that value.

Set this parameter to the minimum allowable stop size to be used in pips.

dblSoftTakeProfitPips, dblSoftStopLossPips

These two parameters define the size of the *Take-Profit* and *Stop-Loss* to be used in the *Market Orders* placed. Please note that this expert advisor, places orders in two steps. First it places the order without a *Take-Profit* and *Stop-Loss* and then subsequently, after knowing the exact opening price, updates it with the correct stops so as to mitigate the effects of spread and slippage.

Set these parameters to the stop sizes of the *Take-Profit* and *Stop-Loss* in pips.

dblSoftTrailStopPips, dblSoftTrailStopPips

The trailing functionality applied to open orders is only in effect when in profit and never while it still in loss after the initial placement. These two parameters define how the trailing is applied to the orders.

The *Trailing Step* parameter defines the multiple with which the profit has to move before the *Trailing Stop* is applied. For example if the *Trailing Step* value is 9 pips and the *Trailing Stop* value is 6 pips, then the order profit has to be higher by 9 pips before the stop is advanced so as to be 6 pips less than the step point. So if the profit is 18 pips (multiple of 9 pip step), then the stop is set 6 pips below that, at the 12 pip level.

Alternatively as an example if *Trailing Step* value is 6 pips and the *Trailing Stop* value is 9 pips and the profit is at 24 pips (multiple of 6 pip step), then the stop is set 9 pips below that, at the 15 pip level.

Set these parameters to the sizes of the *Trailing Stop* and *Trailing Step* in pips.

**dblLotsMinimum,
dblLotsMaximum,
dblLotsStep**

The expert advisor applies money management automatically based on fractional risk calculations that adhere to limits set by these parameters with regards to volume allocation, namely *Minimum* and *Maximum Lot Sizes* as well as allowable resolution or the value of *Stepping* between lot sizes. Internally, these parameters are compared to the allowable *Market* values and adjusted accordingly.

Set these parameters accordingly to the limitations you want to set for the money management implementation.

**dblSlippageEnterPips,
dblSlippageExitPips**

When orders are placed (enter) and then later when they are forcefully closed (exit), a certain amount of slippage can be tolerated, but should slippage be above values defined in these parameters, the operations are retried or ultimately cancelled.

During the placement of an order, should the slippage be higher than allowed by the *dblSlippageEnterPips* parameter, the operation is retried until it succeeds within the slippage tolerance or otherwise cancelled after a certain number of retries.

When orders exit at the *Take-Profit* or *Stop-Loss*, slippage cannot directly be limited, but when the expert advisor forcefully closes orders due to expiration or other factors, the slippage limit set by *dblSlippageExitPips* is used and operations will be retried in order to succeed within tolerances or cancelled after a predetermined retry count.

Set these parameters according to your required slippage limits (or tolerance) in pips.

**dblRiskMaximum,
dblMarginMaximum**

When orders are placed, the allocated volume is calculated based on fractional risk money management rules as well as margin exposure defined by these two parameters.

The *Maximum Risk Percentage* is used together with the defined *Stop-Loss* size to calculate the maximum allowable loss and thus the volume (lots) to be used. On the other hand the *Maximum Margin Percentage* is also used to calculate the volume based on the maximum allowable margin, related to your leverage ratio. The final volume allocated to the order is the lesser of these two calculated values, thus respecting both conditions.

Set these parameters according to your money management requirements for maximum loss risk and maximum margin as percentages.

dblBalanceMinimum

During bad spells of multiple losses, you may want to safeguard your account balance against drawdowns. This parameter allows you to prevent the expert advisor from trading anything when your balance falls below the specified *Minimum Balance*. At this point the expert advisor, will no longer trade unless the balance is augmented by a deposit or by gains from other expert advisors or manual trading.

Set this parameter according to your money management requirements for a minimum balance of your account currency.

intMagicMajor, intMagicMinor

As with all *Expert Advisors*, orders are identified by a *Magic Number*, so that they can coexist with other *Expert Advisors* and not have orders altered by running multiple and different *Expert Advisors* on the same account. You can run multiple instances of this *Expert Advisor* against different *Symbols* with the same *Magic Number* without any problems, and for this reason it is recommended that these parameters not be changed. Unless there are very specific reasons for changing these two parameters, they are best left as they are.

Local Variables

intParamUpdateMins

The *Expert Advisor* will read the stored *Global Variables* and refresh the internal values, on every new bar tick but this interval may be too long. For example, on a H1 Chart, this would be every hour. For this reason it also refreshes the data at a set interval defined by this parameter so as to allow less time between the updates.

Set this parameter to the number of **minutes** of the interval that *Global Variables* are read and data refreshed.

intRetryCount, dblSleepTimeMean, dblSleepTimeMax

When orders are placed or updated, the operations may not execute immediately and fail due to several reasons that sometimes occur. In these cases, the *Expert Advisor* retries the operation, sleeping a certain random amount of time between each retry until it is successful or the maximum number of retries is reached.

Set these parameters in accordance to your judgment and conditions in your environment. The sleep time is in units of **seconds**.