Aeterna Strategy Engine — User Manual

Applies to version 1.x. The settings and section names in this manual describe v1 of the engine. A later major version may add, rename or remove settings.

How to use this manual

This manual is laid out to follow the flow of the EA's **Inputs** tab, one section per settings group. Whatever group you are looking at in MetaTrader, there is a matching section here that explains it.

Each settings group gets its own section, opening with a short description of what the group does, followed by any explanation the group needs and then a table of every input in it. Longer groups are broken up with sub-headings so you can jump to the part you need.

Three text styles are used consistently throughout:

Style	Means
Bold	the name of a setting, exactly as it appears in the Inputs tab
`Code`	a value you enter or an option you select from a dropdown
Italic	a settings group referred to in passing, or a mathematical symbol

---

Contents

Getting started

What this EA is

Aeterna Strategy Engine (Aetarna) is not a single strategy. It is a strategy creation and trade management framework, aimed at providing the best tools possible for an algorithmic trader. Tools that leverage the inherent strengths of algorithmic trading, I.E. risk management without human error, vast analysis of market data, fast and precise trade execution, complex position management, and scalability via strategy and asset diversification.

Aeterna is heavily optimized in all its features to ensure fastest accurate backtesting and low resource usage for setups running multiple instances. Built-in rolling indicator calculations, a sophisticated position tracking system, a user interface that does not process information unless it is being displayed, and many more optimizations all add to efficiency.

Built to adapt. Markets do not hold still. Volatility regimes shift, ranges become trends, correlations break, and a configuration that performed well for months can quietly stop matching the market it was built for. Aeterna is designed around that reality: instead of one strategy to trust indefinitely, it gives you a toolkit to reshape as conditions change — swap a signal, tighten a filter, move the whole configuration to a different volatility scale — without rebuilding anything or abandoning what already works.

Version 1 is the first iteration of that toolkit, and it is built to keep growing. The architecture is deliberately modular: every signal, filter and exit is an independent module reading a shared core, so new tools can arrive with each update without disturbing the configurations already running. As the toolkit expands, so does what you can adapt to — and this manual grows with it.

Input sections are nested with '>'. An input name like *Entry Gates* > *Volatility* > *Scope 1* is a subsection of *Entry Gates* > *Volatility*, itself a subsection of *Entry Gates*.

Every module has its own Enable. A parent's Enable gates its children — a sub-feature acts only when both its own Enable and its parent's are switched on. Nothing is active by implication.

Requirements

Platform	MetaTrader 5
Account type	**Hedging** required for more than one position per side and for Buy and Sell (Hedge) direction mode. Netting accounts work only in single-position configurations.
Symbols	Any. The only feature that reads symbols other than the chart's is the Correlated Asset Shock detector, and the symbols it watches are configurable.
Timeframe	Any. Individual modules carry their own timeframe inputs and are independent of the chart's.

Hosting	A VPS is recommended. The engine is tick-driven and its protection features can only act while the terminal is running.
Instances per chart	Multiple instances may run on the same symbol, each on its own chart.
Magic number	Each instance requires a unique magic number. The engine only ever manages positions carrying its own magic number.

How the engine processes a tick

MetaTrader calls the EA once for every tick — every new price update it receives for the chart's symbol. Each call runs from start to finish before the next tick arrives, so nothing overlaps: whatever the EA decides on one price update is fully resolved before it looks at the next. That also means order matters — everything below happens in a fixed sequence within a single tick, and an earlier step can end the process early, before later steps get a chance to run.

1. **Invalid settings** — if a setting failed validation when the EA loaded, it stays on the chart but does nothing beyond refreshing the GUI, so you can see what needs fixing.
2. **A close-all in progress** — if an account-wide or this-EA-wide flatten is already under way, it is retried and the tick ends there. A one-sided (Buy or Sell) close in progress does not end the tick — the other side keeps trading normally while it finishes.
3. **Position bookkeeping** — the EA's record of its own open positions is brought in line with the terminal.
4. **Drawdown protection** — the day's and the open positions' drawdown are checked, and acted on if a limit is breached. A daily lockout or an open-position cooldown ends the tick here.
5. **News protection** — the scheduled calendar close action runs here, if configured. Blocking new entries around news is handled later, during entry evaluation.
6. **Market data and indicators** — prices and every enabled indicator, tracked levels, and on-chart overlay are refreshed.
7. **Protection stop check** — if a protection feature has already shut trading down, the tick ends here. Data and the GUI keep updating — only trading stops.
8. **Market Anomaly** — the detectors are checked and the circuit breaker decides. While halted, new entries are always blocked; existing positions keep being managed only if the halt did not close them.
9. **Trade plan** — entry gates, signal checks, and stop/target levels are worked out.
10. **Risk sizing** — the lot size is applied to the plan.
11. **Execution** — the order is sent, if the plan calls for one.
12. **Position management** — trailing, break-even and basket rules run on positions already open.
13. **GUI refresh** — only the tab currently in view.

Two things follow from this order: protection always takes priority over new trading, and the manual controls are checked after the protection guards — so using *Resume* by hand can never override an active protection stop.

Distance modes

Most distance inputs in the EA are a **Mode** plus a **Value**, and the mode set is consistent:

Mode	Meaning
Points	A fixed number of points. `value × _Point`.
ATR Multiple	A multiple of the **Trade ATR** (see Trade ATR, further in this manual), not of any module-local ATR.
Percent of Price	A percentage of the relevant price — the entry price, the level's own price, or the anchor, depending on the feature.

Where a feature adds a mode beyond these three, that mode is described in the feature's own section (for example the Spread gate's ratio modes and the Take Profit's risk-to-reward mode).

Because ATR mode resolves against the single Trade ATR, changing the Trade ATR timeframe or period rescales every ATR-mode distance in the EA at once. This is deliberate — it lets one setting move the whole configuration to a different volatility scale — but it means the Trade ATR is not a casual setting.

The on-chart GUI

For an algorithmic trader running several configurations at once, the GUI is the difference between managing by feel and managing by evidence. Every tab reports the same live state the EA is trading on.

At a glance it answers the questions that matter while the EA is running: is it actually configured correctly, is anything currently stopping it from trading, what does each signal module think right now, what levels or ranges is it watching, how much is currently at risk across every open position, and what has it actually closed out over time. All of that without opening a log file or the Inputs dialog — and it holds for every instance you run, since each chart carries its own self-contained GUI.

- **EA Status** — reports any setting that failed validation, naming the exact group, field and reason, so a misconfiguration is visible immediately instead of discovered later as a silently inactive module.
- **Manual Controls** — the EA's only write surface: pause or disable new entries and position management independently, and flatten positions on request.
- **Entry Gates** — every configured entry gate with a live pass/block reading, so you can see exactly which condition is holding a trade back right now.
- **Directional Bias** — live readings from every enabled Trend, Extension and Momentum tool, and whether they currently agree on a direction.
- **Levels** — tracked swing levels: how many are valid, broken or pending, and which ones price is currently approaching, testing or penetrating.
- **Risk** — the configured lot sizing and drawdown limits, plus a live read of total exposure across every open position.
- **Positions** — every position this instance currently has open, with side, size, entry price and running P&L.
- **Position Management** — the configured exit and basket rules across single-position, multi-position and macro scopes, plus a live read-out of any active basket's cycle P&L.

- **News Events** — the configured calendar filter and the upcoming matched events it is currently watching for.
- **Results** — realized closed-trade performance broken down by day, week, month and since inception.
- **System Flags** — the runtime state currently in force — lockouts, halts, cooldowns — distinct from input validation and from the entry gates.

Settings reference

The sections below appear in the same order as the groups in the EA's Inputs tab.

EA Required Settings

The few settings every instance needs regardless of which modules are enabled — how the EA identifies its own positions, an optional label for telling multiple instances apart, and the entry order's slippage tolerance.

Input	Purpose
Unique magic number	Identifies this instance's positions. Must differ between instances. The engine never touches positions that are not its own, except where a Close Scope is explicitly set to the whole account.
Strategy Name	Display only. Shown in the GUI title bar and as its own section on the Status tab, so a profile with several instances is readable at a glance. No effect on trading.
Max Deviation (Points)	Slippage tolerance the broker may fill an **entry** market order within. Closes and modifies are deliberately unconstrained so exits fill regardless of price movement. Too tight invites requotes and rejections on instant-execution brokers; too loose accepts worse fills. `0` accepts only an exact-price fill and is the most rejection-prone setting.

GUI

Controls for the on-chart GUI described earlier — whether it is drawn at all, which tab it opens on, and a calibration escape hatch for its text-measurement layout engine.

Input	Purpose
Enable	Master switch for the on-chart GUI. Turning it off removes all GUI drawing cost.
Default Tab	Which tab is shown when the EA loads.
Layout Width (%)	Calibration for the layout engine's text measurement, which drives every column position and the GUI's own size. `100` trusts the measurement. Raise it if column text visibly overlaps on your machine, lower it if the layout looks too loose. Changing it reinitialises the EA.

Trade ATR (Entries & Exits)

The single Average True Range reading every ATR-Multiple distance in the EA resolves against — stop losses, take profits, trailing distances, volatility gates, and any other Mode/Value pair set to ATR Multiple all read from

this one calculation. There is no per-feature ATR: changing the timeframe or period here rescales every ATR-based distance in the configuration at once (see Distance modes).

Input	Purpose
ATR Timeframe	The chart timeframe the ATR is calculated on. Independent of the timeframe the EA's own chart happens to be running on.
ATR Period	The number of bars the ATR averages over.

Levels > Swing

Tracks recurring price pivots — swing highs and lows — as reference data only: nothing here places or manages a trade by itself. Other modules (entries, exits, filters) read a tracked level's state independently, so this group must be enabled for anything downstream that depends on it to have data.

How a level is detected. A pivot is found with a neighbor-bar test — a bar whose high (for a swing high) or low (for a swing low) stands out against the bars around it. The EA then measures how far price travelled on each side of that pivot, and the level is only tracked if both of those ranges fall within the Minimum and Maximum Range settings. Ranges are measured as a percentage of the level's own price.

A level is only as strong as its **smaller** range — a long run into a level followed by only a small move away from it is not a genuine swing, so both sides have to qualify on their own.

Level states. Every tracked level sits in one of four states, counted on the Levels tab:

- **Pending** — the range before the level qualified, but the range after it has not developed far enough yet. Not yet tradeable.
- **Valid** — both ranges qualified. Drawn on the chart and tracked.
- **Broken** — price pushed past the level by the Break Distance. Still tracked and drawn, now watched for a reclaim. The price interaction and trade expectation directions with this level are now reversed.
- **Invalid** — price came back through a Broken level by the Invalidation Distance. The level is dead and no longer used.

Only a reclaim retires a level for good — being broken does not. A level that runs away and never returns to test itself stays Valid indefinitely.

The interaction system. Interaction is a live read of how close price currently sits to a Valid or Broken level, reported in three non-overlapping bands measured outward from the level, on the side price approaches it from (above a Valid low or a Broken high; below a Valid high or a Broken low):

- **Approaching** — closing in on the level, within the configured Approaching Range.
- **Testing** — inside a narrow band straddling the level itself (its own Test Range on both sides).
- **Penetrating** — has pushed through the level, within the configured Penetrating Range on the far side.

Price outside all three bands, in either direction, registers no interaction with that level.

All four band widths share one Mode (points, ATR multiple, or percent of the level's own price) and are configured independently, so any one band can be switched off by setting its own value to zero. Interaction is

a live reading — it is never drawn as a fixed line, only its optional range overlay is, since it changes constantly with price.

Input	Purpose
Enable	Master switch. Must be on for any consumer (entries, exits, filters) to have level data.
Timeframe	The chart timeframe pivots are detected on.
Minimum Range of Levels (%)	The tracking floor — both the range before and the range after a candidate must clear this percentage of the level's own price before it becomes Valid.
Maximum Range of Levels (%, 0=Off)	An optional ceiling — a level drops out of tracking if its still-growing range after it outgrows this. `0` = no ceiling.
Break Mode	Unit (points or percent of the level's own price) for the Break Distance below.
Break Distance (Pts/%, 0=Immediate)	How far past a Valid level price must push to mark it Broken. `0` = broken on the bare wick-through.
Invalidation Mode	Unit (points or percent of the level's own price) for the Invalidation Distance below.
Invalidation Distance (Pts/%, 0=Immediate)	How far back through a Broken level price must reclaim to retire it for good. `0` = invalidated on any reclaim.
Interaction Mode	The unit (points, ATR multiple, or percent of the level's own price) shared by all four interaction band values below.
Test Range — Approach Side (Pts/ATR/%)	Width of the Testing band on the approach side of the level.
Test Range — Penetrate Side (Pts/ATR/%)	Width of the Testing band on the penetrate side of the level.
Approaching Range (Pts/ATR/%)	How far out the Approaching band extends beyond the Testing band.
Penetrating Range (Pts/ATR/%)	How far past the Testing band the Penetrating band extends.
Display (Draw on Chart)	Draws tracked levels on the chart.
Swing High Color	Chart color for swing high levels.
Swing Low Color	Chart color for swing low levels.
Display: Interaction Ranges	Which levels get their interaction bands drawn: `Off`, `Current` (only levels currently engaged), `Nearest` (closest level on each side), or `All`.
Approaching Range Color	Chart color for the Approaching band, when Display: Interaction Ranges is on.

Input	Purpose
Testing Range Color	Chart color for the Testing band, when Display: Interaction Ranges is on.
Penetrating Range Color	Chart color for the Penetrating band, when Display: Interaction Ranges is on.

Entry Gates

Entry gates decide whether a trade is allowed to be taken at all. They do not generate signals — that is the Entry group's job — they only permit or block. **Every enabled gate must pass** before an entry is placed.

Gates fall into two kinds:

- **Condition gates** — Trade Times, News, Volatility, Regime. These depend only on the clock and on market conditions, never on a signal, so they can block before any signal is even considered.
- **Signal gates** — Direction, Initial Position and Additional Positions. These depend on the proposed trade itself, so they are only checked once a signal exists.

Entry Gates > Direction

Which side the EA may trade. This is the most fundamental gate, so it is checked before any condition-based one.

Input	Purpose
Mode	`Buy Only` / `Sell Only` — one side only. `Buy or Sell (One Direction at a time)` — either side, but never both open at once. `Buy and Sell (Hedge)` — both sides may be open simultaneously; requires a hedging account.

Entry Gates > Trade Times

Restricts trading to a daily time window and a chosen set of weekdays, and optionally closes positions when that window ends. All times are in **broker server time**, not your local time.

The window may cross midnight — a Start of `22:00` with a Stop of `02:00` is treated as a single evening session running into the next day. Setting Start and Stop to the same time disables trading entirely rather than allowing it all day.

Closing out is separate from blocking entries, and comes in two independent forms: a nightly close at the end of each session, and a weekly close before any day the EA is switched off. The weekly one lets the EA hold positions overnight as normal but still go flat ahead of the weekend.

Input	Purpose
Enable	Master switch for the gate. Off = trading is never restricted by time.

Input	Purpose
Start Hour / **Start Minute**	When the daily trading window opens, in server time.
Stop Hour / **Stop Minute**	When the daily trading window closes, in server time. Equal to the start = no trading at all.
Allow Monday … **Allow Sunday**	Per-weekday switches. A day left off is a day the EA does not trade.
Close Positions After Hours	Flattens at the end of each daily session. Off = positions are held overnight.
Close All Before Weekend	An independent weekly flatten that fires on the last session before a day the EA is switched off. Lets you hold overnight yet still be flat over the weekend. Needs the gate enabled with a real (non-equal start/stop) window to have a session edge to fire at; without one it stands down and leaves positions alone.
Allow Additional Entries After Hours	Permits adding to an **existing same-side** position outside the window. Never opens a new side, and overrides only this gate — the weekend close and every other gate still block.

Entry Gates > News

Blocks trading around scheduled economic-calendar events, and can optionally close open positions while an event window is active.

Live trading only. The MetaTrader Strategy Tester does not provide calendar data, so this gate does nothing in a backtest — it neither blocks entries nor closes positions there. A strategy that depends on it will therefore backtest differently from how it trades live.

You choose which events matter with two filters that must **both** match. Headline keywords are matched as case-insensitive fragments of the event's name, and the currency list narrows those to the economies you care about — without it, a keyword like `CPI` would match every country's CPI release. All matched events share the same before/after window and the same action.

Blocking entries and closing positions are separate things. Whenever the gate is enabled and price is inside a window, entries are blocked — there is no separate switch for that. Closing existing positions is an independent choice with its own toggle.

Input	Purpose
Enable (Live Only)	Master switch. Has no effect in the Strategy Tester.
Watched Currencies (CSV, Empty = All)	Comma-separated currency codes, e.g. `USD,EUR`. Leave empty to match every currency.
Watched Headlines (CSV, Name Keywords)	Comma-separated keywords matched against the event name, e.g. `Nonfarm,FOMC,CPI`.

Input	Purpose
Minutes Before Event	How long before a matched event the window opens.
Minutes After Event	How long after a matched event the window stays open.
Close Positions In Window	Flattens the EA's own positions while a window is active. Re-checked continuously, so a position opened mid-window is closed again rather than riding it out.
Allow Additional Entries In Window	Permits adding to an **existing same-side** position during a window. Overrides only this gate; every other active gate still blocks.
Calendar->Server Time Offset (Minutes, If Needed)	Corrects a mismatch between the calendar's timestamps and your broker's server time. Leave at `0` unless windows are visibly landing at the wrong time.
Calendar Scan Window (Days, Look-Ahead & Look-Back)	How many days either side of now the EA scans for matching events.

Entry Gates > Volatility > Scope 1

Restricts trading to a chosen band of market volatility, measured as an ATR.

A scope passes while its ATR sits between its Min and Max thresholds. Either bound can be switched off individually by setting it to `0`, so a scope can enforce a floor only, a ceiling only, or both.

This gate uses its **own** ATR settings, not the shared Trade ATR — changing Trade ATR does not affect it.

If a scope's ATR cannot be read yet — typically not enough history loaded — that scope blocks rather than passes, so the EA never trades on an unknown volatility reading.

Two independent scopes are provided, configured identically. Both must pass, so enabling the second can only narrow trading further — which is what allows a requirement such as short-term volatility elevated while longer-term volatility stays contained.

Input	Purpose
Enable	Switches this scope on. Each scope is enabled separately.
Timeframe	The chart timeframe this scope's ATR is measured on.
ATR Period	The number of bars this scope's ATR averages over.
Value Type (Flat / Points / Percent)	How your thresholds below are interpreted. `Flat (price)` compares directly against the ATR in price terms; `Points` reads them in points; `Percent of Price` reads them as a percentage of current price. There is no ATR-multiple option — the measured value already *is* an ATR.
Min Threshold (0=Off)	Volatility must be at or above this to trade. `0` = no floor.

Input	Purpose
Max Threshold (0=Off)	Volatility must be at or below this to trade. `0` = no ceiling.

Entry Gates > Volatility > Scope 2

A second, independent volatility band with the same settings as Scope 1 above. Both scopes must pass for an entry to be allowed.

Input	Purpose
Enable	Switches this scope on.
Timeframe	The chart timeframe this scope's ATR is measured on.
ATR Period	The number of bars this scope's ATR averages over.
Value Type (Flat / Points / Percent)	As Scope 1.
Min Threshold (0=Off)	Volatility must be at or above this to trade. `0` = no floor.
Max Threshold (0=Off)	Volatility must be at or below this to trade. `0` = no ceiling.

Entry Gates > Regime (Hurst)

Restricts trading to a chosen type of market behaviour — trending, ranging, or anything in between — using the **Hurst exponent**.

What the Hurst exponent measures. Given a stretch of recent price movement, the Hurst exponent (*H*) is a single number describing whether that movement tends to *continue* or to *reverse*:

Reading	Market behaviour	What it means for you
H below 0.5	**Mean-reverting** — moves tend to be given back	Ranging conditions; favours fade and reversion strategies
H around 0.5	**Random** — no memory either way	No exploitable persistence in either direction
H above 0.5	**Trending** — moves tend to continue	Directional conditions; favours trend and breakout strategies

The further from 0.5, the stronger that tendency. *H* is a measure of *behaviour only* — it says nothing about direction, so it never suggests buying or selling, only whether current conditions suit the kind of strategy you have configured.

Setting the band. The gate passes while *H* sits between Min H and Max H. Either bound is switched off with `0`, which is what makes one-sided filters natural:

- Trending only — Min H `0.55`, Max H `0`
- Mean-reverting only — Min H `0`, Max H `0.45`

How it is measured. *H* is estimated by Rescaled Range (R/S), the classic method. The window of recent returns is split into chunks of several different sizes, each size is measured for how far price ranged relative to its own variability, and *H* is the slope of that relationship across sizes. The Sub-period settings define which chunk sizes are used: a wider span sampled at more points gives a steadier, slower-moving reading.

If *H* cannot be computed — not enough history, or a window too flat to produce a slope — the gate blocks rather than passes, so the EA never trades on an unknown regime.

Cost. The full window is re-measured at every chunk size once per bar, when the closing bar is added to the window. Refreshing *H* in between costs very little: inside a bar the only thing that changes is the forming bar's own move, which can fall inside at most one chunk of one size, so only that chunk is re-measured and the rest of the window's measurements are reused. Leaving Update Frequency on every tick is therefore inexpensive, and it keeps the reading tracking the forming bar. Switching to every bar or every few minutes is a choice about how closely *H* follows the forming bar, not a way to save meaningful work. The Window and Sub-period Count still set what that once-per-bar measurement costs.

Input	Purpose
Enable	Master switch for the gate.
Timeframe	The chart timeframe the returns are taken from.
Window (Returns)	How many recent returns *H* is measured over. Larger = steadier but slower to react, and more expensive.
Applied Price	Which price of each bar the returns are built from.
Min H (0=Off)	Trade only while *H* is at or above this. `0` = no floor.
Max H (0=Off)	Trade only while *H* is at or below this. `0` = no ceiling.
Sub-period Min	Smallest chunk size used in the estimate.
Sub-period Max	Largest chunk size used. Cannot exceed the Window — there would be no data to measure.
Sub-period Count (Log-Log Fit Points)	How many chunk sizes are sampled between Min and Max. Needs at least 2 for a slope to exist; more gives a steadier fit at more cost.
Update Frequency	How often *H* is re-calculated: `Every Tick`, `Every Bar`, or `Every N Minutes`.
Update Minutes	The interval used when Update Frequency is set to minutes.
Display Decomposition (Diagnostic Overlay)	Draws the measurement window on the chart, for checking what the gate is actually reading.
Display: Window Bracket Color	Chart color for that overlay.

Entry Gates > Initial Position

Controls when the EA is allowed to open a **first** position on a side, based on how the **previous trade on that side finished**. It applies only when the side is currently flat — once a position is open, adding to it is governed by Additional Positions instead.

What it is for. Back-to-back re-entry is one of the most common ways a configuration bleeds. After a stop-out, price is often still in the move that caused the loss, and a signal firing immediately afterwards tends to be the same failed idea a second time. Equally, after a winner, re-entering instantly can mean buying right at the extreme of the move you just profited from. This gate lets you impose a deliberate cooling-off — in time, in price, or both — and to apply it only in the circumstances you actually care about.

How it decides. Four questions, in order:

1. **Which previous trade do I look at?** Set by *Direction to Gate/Block*, relative to the trade now being proposed — the last trade on the **Same** side, the **Opposite** side, or whichever closed most recently (**Any**).
2. **Does that trade qualify?** Set by *Trigger* — only act if it was a **Win**, only if it was a **Loss**, or **Any** result. If the trade does not match, the gate stands aside and the entry is allowed.
3. **Has enough changed since?** The two sub-gates: *Time Gap* (minutes elapsed since it closed) and *Price Distance* (how far price has moved from it).
4. **How do those combine?** *Combine Mode* — **AND** requires both, **OR** requires either.

The gate passes silently in several situations by design: the side already holds a position, there is no previous trade (a first-ever entry), the previous trade does not match the Trigger, or neither sub-gate is enabled. It only ever restricts a re-entry it has a genuine reason to judge.

The Price Distance rule matters. *Price Move Rule* decides which way price must have moved, measured relative to the **new** trade's direction:

- **Pullback** — price must have moved *against* the new trade by the distance. For a new buy, price must be lower than the reference. This is the "don't chase" setting: wait for a better entry than the last one.
- **Confirmation** — price must have moved *with* the new trade by the distance. For a new buy, price must be higher than the reference. This is the "prove it" setting: only re-enter once the move has resumed in your favour.

Reference Price chooses what that distance is measured from — the previous trade's **entry** or its **exit**. Measuring from the exit is usually what you want after a stop-out (how far has price gone since I was stopped); measuring from the entry compares against where you originally committed.

Worked examples — each row is a complete Trigger / Direction / Rule combination.

Goal	Trigger	Direction	Rule	Effect
Don't immediately re-enter after a loss	`Loss Only`	`Same`	`Pullback`	After a losing buy, a new buy needs price meaningfully lower and/or time elapsed
Don't chase after a win	`Win Only`	`Same`	`Pullback`	Take the next signal only at a better price than the last exit

Goal	Trigger	Direction	Rule	Effect
Require the move to resume after a loss	`Loss Only`	`Same`	`Confirmation`	Re-enter only once price has pushed back in your favour
Space out reversals	`Any`	`Opposite`	—	A new buy is judged against the last sell, useful for stopping rapid flip-flopping

Input	Purpose
Enable	Master switch for the gate.
Trigger (Last Closed Trade Result)	Which previous results the gate acts on: `Win Only`, `Loss Only`, or `Any`. A non-matching result means the gate stands aside.
Direction to Gate/Block (Relative to Trigger)	Which previous trade is used as reference: `Same` side as the proposed trade, `Opposite` side, or `Any` (most recently closed of the two).
Combine Mode (Time/Price)	`AND` = both sub-gates must pass. `OR` = either is enough.
Reference Price	Whether Price Distance measures from the previous trade's `Last Position Entry` or `Last Position Exit`.
Max Time From Previous Entry (Minutes; 0=Off)	An expiry on the whole gate. Once the previous trade is older than this, it stops being treated as a meaningful reference and entries are allowed unconditionally. `0` = never expires.

Entry Gates > Initial Position > Price Distance

The price half of the Initial Position gate — how far price must have moved from the previous trade before a new first position is allowed.

Input	Purpose
Enable	Switches on the price sub-gate.
Mode	`Points`, `ATR Multiple`, or `Percent of Price`.
Value (Pts/ATR/%)	How far price must have moved from the Reference Price.
Price Move Rule (Relative to NEW Trade Direction)	`Pullback` = price must have moved against the new trade. `Confirmation` = price must have moved with it.

Entry Gates > Initial Position > Time Gap

The time half of the Initial Position gate.

Input	Purpose
Min Minutes After Last Close (0=Off)	Minutes that must elapse after the previous trade closed. `0` = the time sub-gate is off.

Entry Gates > Additional Positions

Controls whether the EA may open **further** positions on a side it already holds, and where. This is what lifts a strategy from one trade per signal to a managed multi-position structure — and, equally, what stops an unattended EA from stacking a dozen positions on the same move.

What it is for. A single position expresses a single opinion at a single price. Multiple positions on one side let a strategy express something more nuanced: how conviction should build as a move develops, where exposure should concentrate across a price range, and how much of the position should be committed at any one level. This group defines the rules that structure governs — the direction a stack is allowed to extend, the minimum spacing between entries, how many are permitted, and whether each one needs its own signal.

Those four controls compose rather than select from a fixed menu. Familiar structures fall out of particular combinations — adding only into profit, adding only into adverse movement, accumulating on repeated signals, spacing entries across a range — but they are examples of what the rules produce, not modes you choose between. The intent is that you describe the structure you want and the gate enforces it, including structures that have no conventional name.

The lot size of each additional entry is set separately, under *Risk Management* > *Lot Sizing* > *Additional Position Scaling*, so the **shape** of a stack (defined here) and its **weighting** (defined there) are configured independently.

Mode sets the direction a stack extends. Together with Price Distance, *Mode* decides where an add is permitted relative to the positions already open:

Mode	A new buy is allowed	A new sell is allowed
Average Down	at least the distance **below** the lowest existing buy	at least the distance **above** the highest existing sell
Pyramid	at least the distance **above** the highest existing buy	at least the distance **below** the lowest existing sell
Both	at least the distance away from **every** existing entry, in either direction	as for buys

Note that Average Down and Pyramid anchor to the **extreme** of the existing stack, not to the most recent fill — so a stack can only ever extend in the chosen direction. `Both` instead requires clearance from every open entry, which allows filling a genuine gap between two existing entries while still preventing entries from clustering on top of one another.

Max Positions is the safety limit. It caps positions **per side**, so under Buy and Sell (Hedge) each side gets its own allowance. Left at `1`, the EA takes a single position per side and this whole group is inert. Set to `0`, there is

no cap — meaning position count is then limited only by your Price Distance, Time Gap and margin, so it is worth pairing with a real spacing rule.

Require Signal for Additional Entries decides what drives an add. With it on, each entry needs its own fresh signal as well as clearing the spacing rules, so the stack builds only where the strategy independently agrees. With it off, spacing alone is enough and the structure becomes purely price-driven — entries fill wherever price reaches the configured separation, regardless of whether the original signal still holds.

Input	Purpose
Max Positions Per Side (0=Unlimited)	Maximum simultaneous positions on each side. `1` = single position per side. `0` = no cap.
Require Signal for Additional Entries	On = an add needs a fresh entry signal as well as passing the spacing rules. Off = spacing alone is enough.
Mode	`Average Down`, `Pyramid`, or `Both` — see the table above.
Combine Mode (Price Distance/Time Gap)	`AND` = both sub-gates must pass. `OR` = either is enough.

Entry Gates > Additional Positions > Price Distance

The spacing half of the Additional Positions gate — the minimum separation between entries on the same side, applied per the parent group's Mode.

Input	Purpose
Enable	Switches on the spacing requirement between entries.
Mode	`Points`, `ATR Multiple`, or `Percent of Price`.
Value (Pts/ATR/%)	The minimum separation from the existing entries.

Entry Gates > Additional Positions > Time Gap

The time half of the Additional Positions gate.

Input	Purpose
Min Minutes Between Same-Side Entries (0=Off)	Minimum time between consecutive entries on the same side. `0` = the time sub-gate is off.

Entry Gates > Spread

Blocks entries while the spread is too wide. Spread is a direct cost paid on every trade, and it widens exactly when you least want to be filled — around news, at the session rollover, and in thin liquidity.

The gate compares the current spread against a limit and blocks if the spread is larger. What makes it useful is that the limit does not have to be a fixed number: it can be expressed relative to the trade the EA is about to place, so the same setting stays sensible whether the stop is tight or wide.

Mode	The limit is	Use when
Points	a fixed number of points	you know the acceptable spread for this symbol
ATR Multiple	a multiple of the Trade ATR	you want the limit to scale with volatility
Percent of Price (Entry)	a percentage of the entry price	trading instruments of very different price scales
Max Ratio of Stop Loss Distance	a fraction of the trade's own stop distance	you want spread capped as a share of what you are risking
Max Ratio of Take Profit Distance	a fraction of the trade's own target distance	you want spread capped as a share of what you stand to gain

The two ratio modes are usually the most robust choice: a value of `0.2` means "never pay more than a fifth of the stop distance in spread", which stays meaningful across symbols, timeframes and volatility conditions without retuning.

If a mode's inputs cannot be resolved — the ATR is not ready, or the trade has no stop or target to measure against — the gate blocks rather than passing.

Input	Purpose
Enable	Master switch for the gate.
Mode	How the limit below is interpreted — see the table above.
Value (Pts/ATR/%/Ratio)	The limit itself, in the unit set by Mode. For the ratio modes this is a fraction, so `0.2` = 20% of the stop or target distance.

Entry > Directional Bias

Directional Bias is one of the EA's two entry signal sources (the other being *Entry* > *Levels* > *Swing*, which derives signals from the tracked levels rather than from indicator readings). Where entry gates only permit or block, this is what actually decides **which way** to trade.

It is built from three independent **theories**, each asking a different question about the market:

Theory	Question	Tool
Trend	Which way is price sloping?	Regression Gradient
Extension	How far is price stretched from a reference?	Distance from a basis line
Momentum	How strongly is price pushing?	RSI

Each theory offers **three independent instances**, so a theory can be measured on three timeframes or three configurations at once — for example a Trend reading on M5, H1 and H4 simultaneously.

There is no master Enable. Directional Bias becomes active the moment any single tool beneath it is enabled, and is entirely inactive when none are. This avoids a top-level switch that could contradict the tools below it.

How the readings combine

Every tool produces one of three readings: **buy**, **sell**, or **neutral** (no opinion right now). Those readings are combined at two levels.

Within a theory — its three instances are merged according to that theory's *Confirmation Mode*:

Mode	An enabled instance reading neutral…	Use when
All	**blocks** — every enabled instance must give a direction, and all must agree	you want strict multi-timeframe confluence
Any	**abstains** — it steps aside and lets the others decide	you want three timeframes to be useful rather than near-impossible to satisfy

Between theories — always **all must agree**, and this is deliberately not configurable. If Trend and Momentum are both active, both must point the same way. Confluence between different theories is the entire reason for having separate theories; confluence between three timeframes of the *same* theory is a tuning choice, which is why only the latter is exposed.

Rules worth internalising

A genuine disagreement always yields no signal. Under both Confirmation Modes, if one instance says buy and another says sell, the result is no trade. *Any* widens which instance is allowed to speak; it never permits trading against one that actively disagrees.

Enabled-but-neutral is not the same as disabled. A theory with no tools enabled is excluded entirely — it has no opinion to offer. A theory that *is* enabled but currently reads neutral **blocks**. This is what guarantees that enabling more theories can only ever narrow the signals you take, never widen them. Without that distinction, an enabled theory going quiet would silently let another theory drive the entry alone.

Entry > Directional Bias > Trend

Measures which way price is sloping, using a **regression gradient**.

The regression gradient

Take the last *N* bars and fit a single smooth line through them — the line that sits as close as possible to all of those prices at once. That is a regression line. Now measure how steeply that line is rising or falling **at its most recent end**. That steepness is the gradient.

- Gradient **positive** → the fitted line is rising → bullish

- Gradient **negative** → the fitted line is falling → bearish
- Gradient near **zero** → no meaningful slope → neutral

Two things make this more useful than simply comparing price now against price *N* bars ago:

It uses every bar in the window, not just the endpoints. A single spike at either end can completely reverse an endpoint comparison, while a regression line barely moves — it is fitted against the whole window, so a lone outlier is outvoted by everything around it. This makes the reading substantially steadier than most simple trend measures.

It measures slope, not position. A moving average tells you where price has been on average. A gradient tells you which way that average is *heading*, and how fast. Price can sit well above a rising average or well below a falling one — the gradient is unaffected by that distance, and reports only direction and steepness.

Degree controls the shape of the fitted line:

- **Degree 1 (linear)** — a straight line. The gradient is the overall direction of the whole window. Steady and slow to change.
- **Degree 2 (quadratic)** — a curve, so it can bend. Because the gradient is measured at the newest end, a curve can register that a trend is *decelerating or turning* while a straight line through the same window still reads strongly trending.
- **Higher degrees** — follow the window's shape more closely still, reacting faster but responding more to short-term wiggles.

Choosing the Gradient Measure

The raw slope is in price units per bar, which is a tiny number on FX (a two-point-per-bar trend reads as `0.00002`) and means something completely different on an index or a metal. *Gradient Measure* rescales it so your threshold is a number you can reason about:

Measure	The gradient is expressed as	Comparable across
Price/Bar	raw price change per bar	nothing — specific to this symbol and timeframe
Points/Bar	the same slope in points	the same symbol, easier to type and optimise
Percent/Bar	slope relative to current price	different symbols
Percent/Day	that percentage rescaled to a full day	different symbols **and** different timeframes

Percent/Day is the one to use if you want a threshold that transfers. Because it rescales by how many bars make up a day, a threshold of `0.5` means "half a percent per day" whether the instance is running on M5 or H4. The other measures require retuning whenever you change symbol or timeframe.

Thresholds

Min |*Gradient*| sets how steep the slope must be before the instance reports a direction at all — below it the instance reads neutral. This is what filters out drifting, directionless markets. *Max* |*Gradient*| sets an upper bound, so an instance can be made to stand aside during a near-vertical move that is likely to be over-extended.

Both are read in the unit chosen by Gradient Measure, so changing that measure invalidates any threshold tuned under the previous one.

Input	Purpose
Enable *(section parent)*	Gates all three Regression Gradient instances at once. The section contributes nothing unless at least one instance beneath it is also enabled.
Confirmation Mode *(section parent)*	`All` or `Any` — how this theory's enabled instances combine. Only meaningful with two or more enabled.

Each of the three instances then carries:

Input	Purpose
Enable	Switches this instance on.
Invert Signal	Flips this instance's reading, so a rising slope reads bearish. For counter-trend or fade strategies. A neutral reading stays neutral.
Timeframe	The chart timeframe this instance measures on, independent of the EA's chart.
Applied Price	Which price of each bar the line is fitted through.
Period (Bars to Fit)	How many bars the line is fitted over. Larger = steadier and slower to turn.
Degree (1=Linear, 2=Quadratic, ...)	Shape of the fitted line — see above.
Gradient Measure	The unit the gradient and both thresholds are expressed in — see the table above.
Min |Gradient| (0=Any Non-Zero)	Minimum steepness to report a direction. `0` = any non-zero slope counts.
Max |Gradient| (0=Off)	Maximum steepness. Above it the instance reads neutral. `0` = no upper bound.
Display Regression Line	Draws the fitted line on the chart, so you can see exactly what the instance is measuring.
Regression Line Color	Chart color for that line.

Entry > Directional Bias > Extension

Reports where price currently stands in relation to mean reversion — how far it has stretched from its mean, and whether that stretch is far enough to act on.

The idea

Markets frequently trade in a mean-reverting manner: price moves away from an average, then returns to it. Extension measures that relationship directly. It needs two things — a reference level standing in for the mean, called the **basis**, and a way of judging whether the current distance from it is genuinely unusual, called the **measure**.

What counts as the mean is yours to define, which is why there are four basis types rather than one. Each offers a different answer to the same question: what price is this market currently centred on?

Unless inverted, the signal points toward the basis, on the expectation that a stretched market reverts:

- Price extended **below** the basis → **buy** (stretched down, expected to revert up)
- Price extended **above** the basis → **sell** (stretched up, expected to revert down)

This is the opposite polarity to Trend, which is deliberate — the two theories answer different questions, and requiring both to agree produces a specific and quite selective setup: a market trending one way that has temporarily overstretched in that same direction. If you would rather treat extension as breakout confirmation than as a fade, *Invert Signal* flips the polarity.

Choosing a basis

Basis	The reference is	Reverts around
Moving Average	the average price over the last N bars	a simple rolling mean — the most common definition of "normal"
Regression	a fitted trend line at its newest point	a mean that is itself **sloping**, so a steady trend does not read as permanently extended
VWAP	the volume-weighted average price since an anchor	the price where the most business was actually transacted, not merely the arithmetic middle
Candle Open	the open of the current candle on a chosen timeframe	a fixed session reference, e.g. the day's open on D1

Regression is worth understanding. With a flat moving average, a market in a sustained uptrend sits above its basis continuously and reads as extended the whole way up — which can mean shorting into strength repeatedly. A regression basis tilts with the trend, so extension is measured against where the trend says price *should* be by now, and only a genuine deviation from that path registers. Degree controls the shape: `0` is a plain mean, `1` a straight sloping line, `2` a curve that can bend with a decelerating move.

VWAP anchors to a Day, Week or Month boundary — resetting at each one — or runs as a fixed rolling window (Period). Anchored VWAP is the institutional reference point; the rolling Period variant behaves more like a volume-weighted moving average.

Choosing a measure

The raw distance from the basis is a price number, which means nothing on its own — 50 points is enormous on one instrument and trivial on another, and enormous in a quiet session but ordinary in a volatile one. The measure divides that distance by something meaningful to turn it into a comparable score:

Measure	Distance is divided by	The score reads as
Percent	the basis price itself	"price is 0.8% away from its mean"
ATR	recent Average True Range	"price is 2 normal bar-ranges away" — adapts to current volatility
Standard Deviation	the window's standard deviation	"price is 2 standard deviations away" — a statistical outlier measure

Moving Average basis + Standard Deviation measure is the classic Z-score — the same construction that underlies Bollinger Bands, where a reading of 2 means price sits at the band edge. If you want a familiar starting point, that is it.

Every basis can be combined with every measure — they are independent engines — so combinations like *VWAP basis + Percent measure* ("how far is price from the session's volume-weighted average, in percent") are equally valid.

Thresholds

Min |Extension| is how stretched price must be before the instance reports anything. Below it, the reading is neutral. This is the main tuning dial: at `2.0` with a standard-deviation measure you are waiting for a two-sigma move, which is uncommon by construction.

Max |Extension| is the safety valve, and matters more than it first appears. A move that is *extremely* extended is often not a stretched rubber band but a genuine regime break — a news shock or a breakout that will keep going. Setting a maximum makes the instance stand aside once the reading passes the point where "it must revert" stops being a reasonable assumption. Left at `0` there is no upper bound, and the instance will keep signalling a fade no matter how far price runs.

Both thresholds are read in the unit set by *Measure Type*, so changing the measure invalidates thresholds tuned under the previous one.

Seeing it on the chart

Display Levels draws the basis line plus the threshold bands — the price levels at which the extension score would actually reach your Min (solid) and Max (dotted) settings. This is the fastest way to sanity-check a configuration: if the bands sit far outside anywhere price has recently traded, the instance will never fire, and if they hug price it will fire constantly.

Input	Purpose
Enable *(section parent)*	Gates all three Extension instances at once. The section contributes nothing unless at least one instance beneath it is also enabled.
Confirmation Mode *(section parent)*	`All` or `Any` — how this theory's enabled instances combine. Only meaningful with two or more enabled.

Each of the three instances then carries:

Input	Purpose
Enable	Switches this instance on.
Invert Signal	Flips the polarity, so an upward stretch reads bullish — turning the fade into a breakout confirmation. A neutral reading stays neutral.
Timeframe	The chart timeframe this instance measures on.
Applied Price	Which price of each bar is compared against the basis.
Basis Type	`Moving Average`, `Regression`, `VWAP` or `Candle Open` — see above.
Measure Type	`Percent`, `ATR` or `Standard Deviation` — the unit the score and both thresholds are expressed in.
Min |Extension| (0=Any Non-Zero)	Minimum stretch required to report a direction. `0` = any non-zero distance counts.
Max |Extension| (0=Off)	Upper bound, above which the instance stands aside. `0` = no bound.
Display Levels (Basis + Entry Thresholds)	Draws the basis line and the Min/Max threshold bands.
Basis Line Color	Chart color for the basis line.
Threshold Line Color (Min Solid / Max Dotted)	Chart color for both threshold bands.
Display Bars (Recent Bars to Draw)	How far back the drawn lines extend.

Extension > Basis — only the settings for the selected Basis Type are read; the rest are inert.

Input	Purpose
MA: Period	Bars averaged, when Basis Type is Moving Average.
Regression: Period	Bars fitted, when Basis Type is Regression.
Regression: Degree (0=Mean, 1=Linear, 2=Quadratic, ...)	Shape of the fitted basis line.
VWAP: Anchor Type	`Day`, `Week`, `Month` (reset at each boundary) or `Period` (rolling window).
VWAP: Period (Used Only When Anchor = Period)	Window length for the rolling VWAP variant.
Candle Open: Timeframe	Which candle's open is used as the reference, e.g. D1 for the day's open.

Extension > Measure — only the setting for the selected Measure Type is read.

Input	Purpose
ATR: Period	Bars averaged for the ATR scale, when Measure Type is ATR.
SD: Period	Bars used for the standard deviation, when Measure Type is Standard Deviation.

Entry > Directional Bias > Momentum

Measures the force behind price movement using the **Relative Strength Index (RSI)**.

RSI compares the average size of recent gains against the average size of recent losses, expressing the result on a 0–100 scale. Above 50, gains have been dominating; below 50, losses have. It is one of the most widely used indicators in trading, and it is used here in its straightforward directional sense:

- RSI **above 50** → buying pressure dominant → **buy**
- RSI **below 50** → selling pressure dominant → **sell**

Note this is the midline reading, not the classic overbought/oversold interpretation — a high RSI is treated as strength, not as a reason to fade. If you want the contrarian reading instead, *Invert Signal* provides it.

How this differs from Trend. Trend measures the direction and steepness of price itself — where price has travelled over a window. Momentum measures the conviction behind that travel: the balance of buying against selling pressure. The two can disagree, and that disagreement is informative. A market can still be grinding higher (Trend positive) while each successive push is weaker than the last (Momentum fading toward 50). Because enabled theories must all agree, pairing the two naturally filters out advances that have run out of force.

Thresholds. *Min Distance from 50* sets how far from the midline RSI must sit before the instance reports a direction — at `10.0`, RSI must be above 60 or below 40. This keeps the instance neutral while price chops around the midline. *Max Distance from 50* sets an upper bound, so the instance can stand aside once a move becomes extreme; at `35.0` it stops signalling beyond 85 or below 15.

Input	Purpose
Enable *(section parent)*	Gates all three RSI instances at once. The section contributes nothing unless at least one instance beneath it is also enabled.
Confirmation Mode *(section parent)*	`All` or `Any` — how this theory's enabled instances combine. Only meaningful with two or more enabled.

Each of the three instances then carries:

Input	Purpose
Enable	Switches this instance on.
Invert Signal	Flips the reading, so a high RSI reads bearish — the overbought/oversold interpretation. A neutral reading stays neutral.
Timeframe	The chart timeframe this instance measures on.

Input	Purpose
Period	Bars used in the RSI calculation. `14` is the conventional RSI period.
Applied Price	Which price of each bar the RSI is built from.
Min Distance from 50 (0=Any Side)	How far from the midline RSI must be to report a direction. `0` = any reading off 50 counts.
Max Distance from 50 (0=Off)	Upper bound, beyond which the instance stands aside. `0` = no bound.

Entry > Levels > Swing

The EA's second entry signal source. Where Directional Bias reads indicators, this reads **price interacting with the swing levels** tracked by *Levels* > *Swing*.

That tracking group must be enabled — this section acts on the levels it maintains and has no level data of its own.

Direction and timing

Direction comes from the level; timing comes from the zone. These are two separate decisions, and keeping them apart is the key to configuring this section correctly.

Direction is determined entirely by the level's type and state combined with your *Expectation*:

Level	Expectation	Signal
Valid low	`Rejection`	**Buy** — the low is expected to hold
Valid high	`Rejection`	**Sell** — the high is expected to hold
Valid low	`Break`	**Sell** — the low is expected to give way
Valid high	`Break`	**Buy** — the high is expected to give way
Broken low	`Rejection`	**Sell**
Broken high	`Rejection`	**Buy**
Broken low	`Break`	**Buy**
Broken high	`Break`	**Sell**

A **broken level's role inverts** — price now sits on the opposite side of it, approaching a broken low from below and a broken high from above — so the same Expectation on a broken level trades the opposite way from a valid one. That inversion is automatic; you do not configure it.

Timing is set by *Interaction*, and only timing. It decides at what point in price's approach the entry is permitted to fire, without affecting which way the trade goes.

The interaction zones

The zones are areas, not price levels, and this is the part worth being precise about. Each zone is a band of price surrounding the level, not a single line that price crosses. The signal is live for as long as price remains inside the selected band, and clears the moment price leaves it. The match is exact — choosing `Testing` means the signal exists only while price is in the Testing band, not once it has reached Testing.

The band widths are not set here. They come from the Interaction settings in *Levels* > *Swing* — the tracking group — and are shared by everything that reads them. This makes those settings considerably more consequential than they first appear: they define both what the Levels tab reports and precisely when every level-based entry may fire. Widening the Approaching band, for instance, gives an approach-based entry a longer window and an earlier average fill.

Because a zone is an area rather than a trigger point, it behaves as a **context window**. That distinction matters most in combination with other features:

- With *Additional Positions* set to **Require Signal**, adds are only permitted while the signal is live — that is, only while price is genuinely inside the chosen band. Selecting `Testing` therefore confines an entire scaling structure to the immediate vicinity of the level, rather than letting it continue building as price runs away.
- A narrow band produces something close to a precise trigger; a wide one produces a sustained permission window. Neither is more correct — they express different intentions.
- Because the signal clears on exit from the band, a level that price passes straight through offers only a brief window, while one price hovers around offers a long one.

Choosing which zone to target

Zone	Price is	Typical intent
Approaching	closing in, not yet at the level	Position ahead of a retest — enter into the expected reaction before it happens, accepting that price may not reach the level at all
Testing	at the level, inside the band straddling it	Trade the reaction as it occurs — the most direct reading of "the level is being tested"
Penetrating	beyond the level, in the band on the far side	Trade what happens after price has pushed through — a liquidity sweep, where price briefly runs past a level to trigger orders before reversing, or a genuine breakout continuation

Some worked intentions:

- **Retest entry.** Valid high, `Rejection`, `Approaching` — sell into the swing high before price arrives, on the expectation the level holds. Earliest fill, highest chance the setup never materialises.
- **Liquidity sweep.** Valid low, `Rejection`, `Penetrating` — buy only once price has pushed *below* the swing low. The reasoning is that the dip beneath the level clears out stops and triggers, and the actual reversal begins from below the level rather than at it. Note the Expectation is still `Rejection`: you expect the level to hold in the end, despite the penetration.

- **Breakout continuation.** Valid high, `Break`, `Penetrating` — buy once price has cleared the swing high, treating the move through as genuine rather than a sweep.
- **Failed-break reversal.** Broken high, `Rejection`, `Testing` — the level broke, price has returned to it, and you are now trading it from the other side.

Screening which levels qualify

Min and *Max Range* filter by level strength, using the same measure as the tracking group — the smaller of the ranges either side of the level. This is deliberately a second, tighter screen layered on top of the tracking floor: track many levels so the chart and the Levels tab stay informative, but trade only the strong ones. Both bounds are independent, and `0` disables that side.

One-Shot retires a level from further entries once it has produced a trade, so a single level cannot generate repeated entries each time price revisits it.

Input	Purpose
Enable	Switches on level-based entries. Requires *Levels* > *Swing* to be enabled for level data to exist.
Expectation (Break/Rejection)	Whether you expect the level to give way or to hold — combined with level type and state to set direction.
Level State (Valid/Broken/Both)	Which levels may fire: `Valid Only`, `Broken Only`, or `Valid & Broken`. Remember a broken level trades the opposite way.
Interaction (Approaching/Testing/Penetrating)	Which proximity band the entry fires in. Timing only — it does not affect direction. Band widths come from *Levels* > *Swing*.
Min Range (% of Price, 0=Off)	Minimum level strength required to trade it. `0` = no floor.
Max Range (% of Price, 0=Off)	Maximum level strength. `0` = no ceiling.
Disable level after a trade (One-Shot)	Retires a level once it has produced an entry, preventing repeat trades from the same level.

Exits

Everything that closes a position, or moves the levels that will close it. Exits come in two fundamentally different kinds, and understanding the split explains most of the behaviour in this part of the EA.

Broker-side exits are the stop loss and take profit attached to a position at the moment it opens. They live on the broker's server. Once set, they execute whether or not your terminal is running, your VPS is online, or the EA is even loaded. This is the safety floor.

Engine-side exits are everything the EA does afterward: moving a stop to break even, trailing it, managing a basket of positions together. These require the EA to be running, because the EA is what performs each modification.

The practical consequence: a configuration whose only protection is engine-side is exposed whenever the terminal is not running. The EA guards against the extreme version of this — if your configuration provides no way to ever close a position, it refuses to arm and reports the problem on the Status tab rather than opening trades it cannot exit.

The position management system

Once a position is open, the EA continuously reviews what its stop and target should be and adjusts them as the configured rules require.

A basket is all of this EA's open positions on one side of the market. Every buy it holds forms the buy basket; every sell forms the sell basket. The two are managed entirely separately, so under Buy and Sell (Hedge) a side can have its own merged stop, its own trailing rule and its own profit target without any reference to the other. A basket contains only this instance's positions on this chart's symbol — positions from another EA, another instance, or opened by hand are never part of it.

Management operates at **three scopes**, evaluated from the widest down:

Scope	Acts on	Status
Macro	every position, both sides combined	reserved — not implemented in v1, no effect
Multi (MPM)	one side's basket, managed as a unit	active when *Multi-Position Management* is enabled
Single (PM)	one position on its own	active per feature under *Position Management*

Higher scopes take priority over lower ones, and this is the single most important thing to understand about exits. A position's stop and its target are controlled separately. Once a wider scope sets one of them, narrower scopes stop adjusting it for the rest of that position's life.

In practice, this means:

- If MPM sets a merged stop across a side's basket, per-position trailing stops leave those positions' stops alone. The basket's stop is now the one in force.
- The stop and the target are held **separately**. MPM can control the stop while per-position rules still manage the take profit on the same position.
- Control is also **per side**. MPM taking over the buy basket's stops leaves every sell position under per-position management.
- Control **persists**. Disabling MPM mid-trade does not return those stops to per-position management; positions opened afterward start fresh.

The reason for the hierarchy is coherence: a basket stop calculated across five positions is meaningless if a per-position rule immediately moves one of those five somewhere else.

Positions are re-adopted after a restart. A restart, a terminal crash, a reboot, or reattaching the EA to the chart does not orphan open positions — the EA recognises any position carrying its magic number on its chart symbol and resumes managing it, trailing, break-even and basket rules included, exactly as though it had never stopped. The condition is the magic number: leave it unchanged and the positions are picked back up, change it and the EA will leave them entirely alone.

Only meaningful changes are sent to the broker. A trailing stop would otherwise move by a fraction of a point at a time, producing a constant stream of modifications for no benefit. *Epsilon* sets the minimum change required before the EA sends one, and applies to both per-position and multi-position management.

Manual pausing suspends only the EA's own management. The Manual Controls tab can pause or disable position management independently of entries, which stops trailing, break-even and basket logic. Broker-side stops and targets stay exactly where they are and remain in force, and every protection feature — drawdown limits, the anomaly engine, close-all operations — continues to run regardless, since those sit outside position management entirely.

Exits > Epsilon

The minimum change required before the EA will send a stop or target modification to the broker. Shared by per-position and multi-position management.

Too small and the EA sends near-continuous modifications for negligible movements. Too large and trailing becomes visibly coarse, with the stop jumping in steps rather than following price. Disabling it entirely means every difference, however small, is sent.

Input	Purpose
Enable	On = only send a modification when the change exceeds the threshold below. Off = send on any change at all.
Mode	`Points`, `ATR Multiple`, or `Percent of Price`.
Value (Pts/ATR/%)	The minimum change required. `0` behaves as disabled.

Exits > Basic > SL

The stop loss attached to a position when it opens — a broker-side exit, and the foundation of the exit configuration. Everything under *Position Management* and *Multi-Position Management* modifies a level that starts here.

The stop loss has a second role. Under *Risk Management* > *Lot Sizing* with a risk-based mode, lot size is derived from the distance between entry and stop: a tighter stop produces a larger position for the same money risked. Changing this setting therefore changes trade size, not merely where the trade closes.

Input	Purpose
Enable	Attaches a stop loss at entry.
Mode	`Points`, `ATR Multiple`, or `Percent of Price`.
Value (Pts/ATR/%)	Stop distance from entry, in the unit set by Mode.

Exits > Basic > TP

The take profit attached to a position when it opens.

A fourth mode is available here. Alongside the usual three, `Risk:Reward` expresses the target as a multiple of the stop distance — a value of `3.0` places the target three times as far from entry as the stop, for a 3:1 reward-to-risk ratio. This mode requires a stop loss to measure against.

Input	Purpose
Enable	Attaches a take profit at entry.
Mode	`Points`, `ATR Multiple`, `Percent of Price`, or `Risk:Reward`.
Value (Pts/ATR/%/RR)	Target distance from entry. In `Risk:Reward` mode this is the reward multiple per 1 unit of risk.

Exits > Position Management

Rules that act on **one position at a time**, adjusting its stop, its target, or its size after it has opened. This is the Single scope described in the Exits overview — the narrowest of the three, and the one a wider scope can lock out of a domain.

Four features sit here, each independently enabled. Three of them move a level; the fourth closes part of the position.

Profit-side and loss-side

Feature	Moves the	Triggers on	Purpose
SL to BE	stop	**profit** reaching the trigger	Remove risk once the trade is working
Trailing SL	stop	**profit** reaching the start	Follow price, locking in more as it runs
TP to BE	target	**loss** reaching the trigger	Reduce the target so a recovery can still exit flat

The two stop-side features are the familiar ones: protect a winner. **TP to BE is a recovery tool**, and the less common idea. When a position moves against you the original target may be unrealistically far away, so this pulls the target back to — or near — break-even, letting a partial recovery close the trade flat instead of requiring a full round trip to the original goal. It accepts a smaller outcome in exchange for a much higher chance of reaching it.

Every adjustment only ever tightens. A stop is never moved further from price and a target is never moved further away — each feature can only improve the position's protection or bring its exit closer. This holds even when two features act at once: if both SL to BE and Trailing SL qualify on the same tick, the tighter of the two results is what gets used. You cannot construct a configuration where enabling another feature loosens an existing level.

Trigger and offset are measured from different things. For the break-even features, *Trigger* is measured as a distance from the entry price — how far into profit (or loss) the position must travel — while *Offset* shifts the

resulting level away from exact break-even, always in the profitable direction. A small offset is the practical way to cover spread and commission so that "break-even" is genuinely flat rather than a marginal loss. For Trailing SL, *Start* is the distance from entry that arms the feature, and *Trail* is the gap the stop then maintains behind current price.

Exits > Position Management > SL to BE

Moves the stop loss to break-even once the position is far enough into profit.

Input	Purpose
Enable	Switches the feature on.
Mode	`Points`, `ATR Multiple`, or `Percent of Price` — the unit for both values below.
Trigger Profit Distance (Pts/ATR/%)	How far into profit the position must be before the stop is moved.
Offset from Entry (Profit)(Pts/ATR/%)	How far beyond entry, in the profitable direction, to place the stop. `0` = exactly at entry. A small positive value covers costs so the exit is genuinely flat.

Exits > Position Management > TP to BE

Moves the take profit back toward break-even once the position is far enough into **loss** — the recovery mirror of SL to BE.

Input	Purpose
Enable	Switches the feature on.
Mode	`Points`, `ATR Multiple`, or `Percent of Price` — the unit for both values below.
Trigger Loss Distance (Pts/ATR/%)	How far into loss the position must be before the target is moved.
Offset from Entry (Profit) (Pts/ATR/%)	How far beyond entry, in the profitable direction, to place the new target. `0` = exactly at entry.

Exits > Position Management > Trailing SL

Follows price with the stop once the position is far enough into profit, locking in progressively more as the move extends.

Input	Purpose
Enable	Switches the feature on.
Mode	`Points`, `ATR Multiple`, or `Percent of Price` — the unit for both values below.
Start Distance (Pts/ATR/%)	Profit distance from entry at which trailing begins.
Trail Distance (Pts/ATR/%)	The gap maintained between price and the stop. Tighter captures more of a reversal but is stopped out by ordinary noise sooner.

Exits > Position Management > Partial Close

Exits the position in stages rather than all at once, taking profit at up to three **targets** — a scale-out ladder. A target is one step of that ladder; the term is used throughout the EA and on the Position Management tab, and is distinct from the swing levels of the *Levels* module.

The targets are ratios of the base take profit. TP1 is whatever *Exits* > *Basic* > *TP* is set to, and the TP2 and TP3 ratios multiply that distance — with a base TP of 30 points and a TP2 ratio of `2.0`, the second target sits at 60 points. Re-tuning the base take profit therefore moves the whole ladder with it.

Enabling this changes what the broker-side take profit does. It is pushed out to the **final** active target. TP1 stops being the exit and becomes the first partial, with the earlier targets watched by the EA as virtual levels. Keeping the last target broker-side is deliberate: if the terminal disconnects, the final exit still fires on the broker's server. The trade-off is that the intermediate partials require the EA to be running, exactly like the other engine-side exits.

The last active target always closes whatever remains. This means the close sizes cannot fail to add up — you only set sizes for the targets *before* the last one. With all three targets active, TP1 and TP2 take their configured shares and TP3 takes the remainder; with TP3 off, TP2 takes the remainder and its own size input is ignored.

Min Lot Policy handles a split the broker cannot execute. Brokers enforce a minimum trade volume, and a small position divided three ways may produce chunks below it. Note this is a different problem from the position being too small to open at all — that is governed by *Risk Management* > *Lot Sizing*. Here the trade is viable and only the split does not fit:

Policy	Behaviour
`Reduce Targets`	Drop TP3, then TP2, until every remaining chunk clears the minimum. Risk is unchanged — the surviving final target still takes the remainder.
`Round To Min Lot`	Raise undersized chunks to the broker minimum, with the final target absorbing the difference. Keeps all targets but distorts the requested split. Falls back to `Reduce Targets` if the minimums alone exceed the position.
`Disable Split (Single TP)`	No ladder — the position exits whole at the base TP.

Policy	Behaviour
`Skip Trade`	Do not open the position at all. For when a specific ladder is the strategy and a silently different one is worse than no trade.

If you close part of a position by hand, the ladder detects the changed volume and resumes at the nearest target it can.

Input	Purpose
Enable	Switches the ladder on. Moves the broker TP out to the final active target.
TP 2 Distance Ratio (0=Off)	TP2's distance as a multiple of the base TP distance. `0` disables this target.
TP 3 Distance Ratio (0=Off)	TP3's distance as a multiple of the base TP distance. `0` disables this target.
TP 1 Close Size (% of Initial Volume)	Share of the original position closed at TP1.
TP 2 Close Size (% of Initial Volume)	Share of the original position closed at TP2. Ignored when TP2 is the last active target, since the last target always takes the remainder.
Min Lot Policy	What to do when the chunks do not clear the broker's minimum volume — see the table above.

Exits > Multi-Position Management

Manages a whole **basket** as one unit — all of this EA's open positions on one side of the market. This is the Multi scope from the Exits overview, and it takes priority over per-position management: when an MPM feature acts on a stop or a target, the equivalent Single feature is locked out of that domain for those positions.

The buy and sell baskets are judged and managed entirely independently. Nothing here combines both sides; that is reserved for the future Macro scope.

This group's Enable is a master switch. Every feature below requires it as well as its own Enable — with the master off, nothing in this section acts regardless of individual settings.

Why manage a basket instead of individual positions

Once a side holds several positions at different entry prices, per-position stops and targets stop describing your actual exposure. Each position exits at its own price, so the side unwinds piecemeal and the result depends on the order they happen to trigger in.

Basket management treats the side as the single trade it effectively is. Every position on it receives the **same** stop or target price, derived once from a shared anchor — so the whole side enters and leaves as a unit, and you can reason about one exit rather than five.

The anchor

Every level-setting feature here measures its distance from an **anchor** — a single reference price representing the basket. This is the central choice in MPM configuration, and each option produces materially different behaviour:

Anchor	The reference is	Character
Average Entry (VWAP)	the volume-weighted average of every entry on the side	The balanced choice — treats the basket as one position at its true average cost
Initial Entry (Oldest)	the first position opened in the cycle	Stable; later adds never move the level. The original trade defines the exit
Latest Entry (Newest)	the most recent position opened	Moves with every add, so the level tracks where you are currently committing
Best Profit Move Entry	the entry currently furthest into profit	Anchors to the strongest leg — a tighter, more conservative exit
Worst Profit Move Entry	the entry currently furthest into loss	Anchors to the weakest leg — the most forgiving exit, and the one that keeps a struggling basket alive longest

Each feature carries its own Anchor setting, so a merged stop can measure from the worst entry while a merged target measures from the average — they do not have to agree.

Min Positions

Every feature has a **Min Positions (Per Side)** threshold, and this is what keeps basket management from interfering with ordinary single trades. Below the threshold the feature simply does not act, leaving those positions to per-position management.

Set to 2, a side holding one position is managed normally by *Position Management*, and basket rules engage only once a second position opens. Set to 1, the feature acts on a lone position too, which hands that stop or target to MPM from the first trade.

The features

Three shapes of feature sit here, and they mirror what Single scope offers:

- **Merge SL / Merge TP** — place one stop or target for the whole side, at a fixed distance from the anchor. These have no trigger; they apply as soon as Min Positions is met.
- **SL to BE / TP to BE / Trailing SL** — the same behaviours documented under *Position Management*, but computed from the anchor and applied to every position on the side. Their triggers and offsets are measured from the anchor rather than from an individual entry.
- **Basket Profit Target** — closes the entire side on money, not on a price level.

Input	Purpose
Enable	Master switch for every MPM feature below. Off = no basket management at all.

Exits > Multi-Position Management > Merge SL

Places a single stop loss for the whole side, at a fixed distance from the anchor.

Input	Purpose
Enable	Switches the feature on.
Min Positions (Per Side)	Positions required on the side before this acts.
Anchor	The reference price the distance is measured from — see the anchor table above.
Mode	`Points`, `ATR Multiple`, or `Percent of Price`.
Value (Pts/ATR/%)	Distance from the anchor to the merged stop.

Exits > Multi-Position Management > Merge TP

Places a single take profit for the whole side, at a fixed distance from the anchor.

Input	Purpose
Enable	Switches the feature on.
Min Positions (Per Side)	Positions required on the side before this acts.
Anchor	The reference price the distance is measured from.
Mode	`Points`, `ATR Multiple`, or `Percent of Price`.
Value (Pts/ATR/%)	Distance from the anchor to the merged target.

Exits > Multi-Position Management > SL to BE

Moves the whole side's stop to the anchor once the basket is far enough into profit — the basket equivalent of the per-position feature, with break-even meaning the anchor rather than any single entry.

Input	Purpose
Enable	Switches the feature on.
Min Positions (Per Side)	Positions required on the side before this acts.

Input	Purpose
Anchor	The reference price, and the level the stop moves to.
Mode	`Points`, `ATR Multiple`, or `Percent of Price`.
Trigger Profit Distance (Pts/ATR/%)	How far past the anchor, in profit, the basket must travel before the stop moves.
Offset from Anchor (Profit) (Pts/ATR/%)	How far beyond the anchor, in the profitable direction, to place the stop. `0` = exactly at the anchor.

Exits > Multi-Position Management > TP to BE

Pulls the whole side's target back to the anchor once the basket is far enough into **loss** — the basket recovery tool.

Input	Purpose
Enable	Switches the feature on.
Min Positions (Per Side)	Positions required on the side before this acts.
Anchor	The reference price, and the level the target moves to.
Mode	`Points`, `ATR Multiple`, or `Percent of Price`.
Trigger Loss Distance (Pts/ATR/%)	How far past the anchor, in loss, the basket must travel before the target moves.
Offset from Anchor (Profit) (Pts/ATR/%)	How far beyond the anchor, in the profitable direction, to place the target. `0` = exactly at the anchor.

Exits > Multi-Position Management > Trailing SL

Trails a single stop for the whole side once the basket is far enough into profit.

Input	Purpose
Enable	Switches the feature on.
Min Positions (Per Side)	Positions required on the side before this acts.
Anchor	The reference price the start distance is measured from.
Mode	`Points`, `ATR Multiple`, or `Percent of Price`.
Start Distance (Profit)(Pts/ATR/%)	Profit distance from the anchor at which trailing begins.
Trail Distance (Pts/ATR/%)	The gap maintained between price and the side's stop.

Exits > Multi-Position Management > Basket Profit Target

Closes a side's entire basket on **money**, rather than at a price level. This is the one feature here that does not set a stop or target — it monitors the side's running profit and closes everything when it reaches your figure.

It counts profit already banked during the cycle, not just what is floating. A cycle begins when a side opens its first position and ends when the side goes flat. Anything realised along the way — a partial close, or a leg that hit its own target while others stayed open — is added to the floating profit of the positions still open. Without this, scaling out would quietly reset progress toward the target every time a leg closed.

Include Costs decides what "profit" means. Off, the measure is price movement only. On, swap and commission are folded in, so the figure reflects what actually reaches the account — which matters most on baskets held long enough for swap to accumulate.

Percent mode is measured against the balance at cycle start, not the live balance — the account balance captured when the side opened its first position. This keeps the goalposts still for the life of the cycle rather than drifting as other trades settle.

Both the target and the loss limit are independently disabled with `0`, so the feature can act as a profit target only, a basket stop only, or both.

Input	Purpose
Enable	Switches the feature on.
Min Positions (Per Side)	Positions required on the side before this acts.
Mode	`Flat Money` = the values below are account currency. `Percent of Balance` = a percentage of the balance at cycle start.
Profit Target (Money or %, 0=Off)	Close the side once cycle profit reaches this. `0` = no profit target.
Basket Loss Limit (Money or %, 0=Off)	Close the side once cycle loss reaches this. `0` = no loss limit.
Include Costs (Swap + Commission in P/L)	Off = price movement only. On = net of swap and commission.

Risk Management

Two distinct jobs sit under this heading, and it is worth separating them clearly because they work at opposite ends of a trade.

Lot Sizing decides how large each position is *before* it opens. **Drawdown Limits** decide when the EA stops trading and closes what is open, *after* losses have accumulated. The first governs how much any single trade can cost you; the second governs how much a sequence of them can.

Neither depends on the other, and a complete configuration usually wants both: sizing alone does not stop a losing run, and drawdown limits alone do not prevent an oversized trade.

Risk Management > Lot Sizing

How position size is calculated. Two modes are available, and they answer different questions.

Mode	Sizes from	Question it answers
Risk by Stop Loss	the distance to your stop, and a risk budget	"How many lots put exactly this much money at risk?"
Lots per Balance (Sample)	account balance alone	"How many lots per unit of account size?"

Risk by Stop Loss is the more disciplined of the two, because it holds the *money* at risk constant rather than the lot size. A wide stop produces a smaller position and a tight stop a larger one, so every trade risks the same amount regardless of where its stop happens to sit. This is also why the Basic stop loss changes trade size: the stop distance is an input to the calculation, not just an exit level.

It follows that this mode **requires a stop loss**. With no stop there is no distance to size against.

Lots per Balance (Sample) ignores the stop entirely and scales purely with account size — for example `0.01` lots per `1000` of balance. Simpler and predictable, but the money at risk then varies with every change in stop distance.

Input	Purpose
Mode	`Risk by Stop Loss` or `Lots per Balance (Sample)`.
Min Lot Policy	What to do when the calculated size falls below the broker's minimum: `Skip Trade` takes no trade; `Clamp To Min Lot` opens at the minimum instead.
Max Lots Cap (0=Off)	Hard ceiling on the calculated size, whatever the mode produces. `0` = no cap.

Min Lot Policy deserves thought. `Clamp To Min Lot` guarantees the trade is taken, but it opens a position *larger* than your risk budget allows — the one place in the EA where a configured risk figure can be exceeded. `Skip Trade` preserves the budget exactly and is the safer choice, at the cost of silently skipping trades on small accounts or very wide stops.

Risk Management > Lot Sizing > Risk by SL

The risk budget — how much money a single trade is allowed to lose — used only when Mode is `Risk by Stop Loss`.

Input	Purpose
Mode	`Flat Money` = a fixed account-currency amount. `Percent of Balance` = a percentage of current balance. `Percent of Max Drawdown` = a percentage of a prop-account allowance, configured below.

Input	Purpose
Value (Money or %)	The budget itself, in the unit set by Mode.

`Percent of Balance` compounds naturally: as the balance grows the money risked grows with it, and as it shrinks the risk shrinks too. `Flat Money` keeps the figure fixed regardless of account performance.

Risk Management > Lot Sizing > Risk by SL > % of Max Drawdown

Prop-account sizing, read only when the risk budget Mode is `Percent of Max Drawdown`.

This sizes against your remaining buffer, not your balance. A prop account has a fixed drawdown floor which ends the account if breached. What matters for sizing is not how much money is in the account but how much is left before hitting that floor:

```
floor            = Initial Balance × (1 − Max Drawdown% ÷ 100)
remaining buffer = current equity − floor
```

Your risk budget percentage is then applied to that **remaining buffer**. The practical effect is that position sizes shrink automatically as the account approaches its limit — the closer you get to failing, the smaller the trades become, which is exactly the behaviour a drawdown rule demands. As equity recovers, sizing expands again.

If equity reaches or passes the floor the buffer is zero or negative, the budget is zero, and no trade is taken.

Input	Purpose
Initial Account Balance	The starting balance the prop firm's drawdown allowance is measured from.
Max Drawdown (% of Initial)	The firm's maximum drawdown allowance, as a percentage of that initial balance.

Risk Management > Lot Sizing > Balance Scaled (Sample)

The sizing table for `Lots per Balance (Sample)` mode. Read only in that mode.

Input	Purpose
Lots per Sample Amount	Lots to trade for each Sample Amount of balance.
Sample Amount	The balance unit the lot figure applies to.

At `0.01` lots per `1000`, a balance of 25,000 produces 0.25 lots.

Risk Management > Lot Sizing > Additional Position Scaling

Adjusts the size of *additional* positions on a side, so a scaling structure does not have to use uniform lots. This is the weighting half of multi-position strategies; the shape is set under *Entry Gates* > *Additional Positions*.

Scaling always applies to the **base** lot size — whatever the sizing mode calculated for the first position — never compounding off the previous position's adjusted size. Both a multiply and an add factor are available, and they combine:

```
lots = base × (Multiply Ratio ^ k) + base × Add Ratio × k
```

where `k` counts from 1 for the first position that scaling applies to, incrementing with each further add.

Setting	Effect
Multiply `1.0`, Add `0.0`	No scaling — every add is the base size
Multiply `2.0`, Add `0.0`	Doubling — base, then 2×, 4×, 8×
Multiply `1.0`, Add `0.5`	Linear growth — base, then 1.5×, 2×, 2.5×
Multiply `0.5`, Add `0.0`	Halving — each add is smaller, front-loading the position

Multiplying ratios above 1.0 compound quickly. A doubling ladder reaches 8× the base size by the third add, and combined with an averaging-down structure that means the largest position sits at the worst price. That is the classic martingale failure mode. The Max Lots Cap above and the drawdown limits below are the guards worth having in place before using one.

Input	Purpose
Enable	Switches scaling on. Off = every additional position uses the base size.
Start From Position (Per Side)	The position number scaling begins at. `2` leaves the first position at base size and scales from the second onward.
Multiply Ratio	Geometric factor applied per scaled position. `1.0` = no multiplication.
Add Ratio	Linear factor, as a fraction of base lots added per scaled position. `0.0` = no addition.

Risk Management > Drawdown Limits

Automatic protection that closes positions and stops trading when losses reach a limit. This is the EA's account-preservation layer, and it sits outside position management entirely — pausing management by hand does not disable it.

This group's Enable is a master switch, exactly like Multi-Position Management. Each scope below also has its own Enable, and both must be on for that scope to act.

Two scopes are provided, measuring different things:

Scope	Measures	Baseline	Resets
Open Positions	drawdown while trades are open	the equity peak reached since the current positions opened	when the side goes flat
Daily	drawdown across the trading day	the higher of balance or equity at the start of the server day	at each server-day rollover

Open Positions protects against a single episode going badly — it starts measuring when positions open, tracks the highest equity reached, and acts when equity falls far enough from that peak. Daily protects against a bad *day* regardless of how many trades produced it, and its lockout persists until the next server day.

Close Scope decides what gets closed on a breach, and the distinction matters on shared accounts. `EA Magic Positions Only` closes just this instance's trades, leaving everything else untouched. `All Account Positions` closes everything on the account, including other EAs' and manual trades — appropriate when the limit represents an account-wide rule such as a prop firm's, and dangerous otherwise.

Risk Management > DD Limits > Open Positions

Input	Purpose
Enable	Switches this scope on. Also requires the Drawdown Limits master Enable.
Mode	`Percent of Peak` = a percentage below the tracked equity peak. `Flat Money` = a fixed currency drop from it.
Value (Money or %)	The drawdown that triggers a breach.
Cooldown After Breach (Minutes; 0=Off)	How long trading stays blocked after a breach before resuming automatically. `0` = no cooldown, trading may resume immediately.
Close Scope	`EA Magic Positions Only` or `All Account Positions`.
Disable EA on Breach	Hard stop — trading ends for good and only reloading the EA resumes it. The EA stays loaded so the Status tab keeps showing what happened and when.

Cooldown and Disable EA are two strengths of the same idea. Cooldown is the soft, self-recovering option; Disable EA is the hard stop requiring a human decision to resume. Using neither means the EA may re-enter almost immediately after a breach, which often defeats the purpose of having the limit.

Risk Management > DD Limits > Daily

Input	Purpose
Enable	Switches this scope on. Also requires the Drawdown Limits master Enable.

Input	Purpose
Mode	`Percent of Peak` = a percentage below the day's starting baseline. `Flat Money` = a fixed currency loss for the day.
Value (Money or %)	The daily loss that triggers a breach.
Close Scope	`EA Magic Positions Only` or `All Account Positions`.
Disable EA on Breach	Escalates the automatic daily lockout to a permanent stop requiring a reload.

The daily lockout is automatic and needs no cooldown setting. Once breached, trading is blocked until the server day rolls over — which is why this scope has no Cooldown input while Open Positions does. Disable EA on Breach escalates that further, to stopped-until-reloaded, preserving the cause on the Status tab for review.

The daily baseline is the higher of balance or equity at the day's start. Taking the higher of the two means an open floating profit at rollover cannot inflate the allowance, so the limit stays honest across positions held overnight.

Market Anomaly

Circuit breakers that stop trading and close positions when the market stops behaving like the market the strategy was built for.

Why an algorithmic system needs this

An automated strategy is a set of rules calibrated against a particular kind of market. Your thresholds, distances and filters all encode assumptions about how price normally moves — how far it travels in an hour, how volatile a typical bar is, how instruments relate to one another. Those assumptions are what make the rules meaningful.

When something extraordinary happens — a central bank intervention, a flash crash, a liquidity vacuum — those assumptions stop holding. The rules keep executing, because rules always do, but they are now being applied to conditions they were never tested against. A stop distance calibrated for normal volatility is meaningless in a move ten times that size. Spread widens past anything the configuration anticipated. Correlations that held for years invert within minutes.

The honest response is not a better rule. It is to recognise that the context has broken and **stop trading until it is restored**. That is what this module does: it watches for conditions that indicate the market is no longer the one your configuration describes, and when it finds them, it gets out of the way.

This is fundamentally different from an entry gate. A gate says "conditions are not right for a trade". A circuit breaker says "conditions are not right for this system to be operating at all".

How it is built

The module separates **measurement** from **response**, which is why the settings are laid out the way they are.

Detectors measure and nothing else. Each watches for one kind of abnormality and reports whether it currently sees one. Each has its own settings for what to measure and what counts as abnormal, and each can be enabled independently.

One shared engine owns everything that happens when a detector trips: whether positions are closed, what gets closed, how long trading stays halted, and when it resumes. The response is deliberately identical no matter which detector fired — an anomaly is an anomaly, and the account-preservation action does not depend on which measurement noticed it.

The consequence for configuration: detector groups answer *what is abnormal*, and the parent group answers *what to do about it*.

What a halt does

A halt disables the EA **internally**. It stays loaded on the chart, the GUI keeps reporting, and in Auto mode it keeps watching for conditions to normalise — it simply stops opening trades. The EA never removes itself: MetaTrader exposes the terminal's Algo Trading switch as read-only, so an unload would be a one-way door that nothing inside the EA could reopen.

Whether existing positions are closed is a separate decision from whether trading halts. With *Close Positions on Trip* off, the halt blocks new entries while leaving open positions running and still managed — appropriate when your exits are the protection and you simply do not want to be adding into chaos.

The response policy

Halt Mode is the key choice. `Auto Re-arm` is for events you expect the EA to sit out and then resume after — a volatility spike that passes. `Hard Halt` is for events serious enough that a human should look before trading continues; it stays halted until the EA is reloaded.

Auto Re-arm requires both timers to be satisfied before trading resumes: the measure must have fallen back below its re-arm threshold and stayed there for the Cooldown, *and* at least the Minimum Halt must have elapsed since the trip. Minimum Halt exists because some event classes arrive in waves — the measure settles, then a second wave follows — and a cooldown alone would resume trading directly into it.

Input	Purpose
Enable Market Anomaly Protection	Master switch. Off = no detector acts, whatever its own settings.
Close Positions on Trip	On = flatten on a trip. Off = halt entries only, leaving open positions running and still managed.
Close Scope	`EA Magic Positions Only` or `All Account Positions`. Applies only when Close Positions is on.
Halt Mode	`Auto Re-arm (resume when normalised)` or `Hard Halt (stay halted until reload)`.
Re-arm Cooldown (Minutes; Auto Mode)	How long the measure must stay normal before trading resumes.

Input	Purpose
Minimum Halt (Minutes; Auto Mode; 0=Off)	Minimum time halted regardless of how quickly conditions normalise. `0` = no minimum.

Market Anomaly > Volatility

Trips when the chart symbol's volatility exceeds a threshold — the simplest form of "this is not the market I was configured for".

This is a one-sided **upper** test: only abnormally high volatility trips it. That distinguishes it from the Volatility *entry gate*, which is a band with both a floor and a ceiling and only blocks entries. This one halts the system.

The re-arm threshold provides hysteresis. Leaving it at `0` re-arms at the same level that tripped, which means volatility hovering around the threshold can trip and re-arm repeatedly. Setting it lower than the trip level requires a genuine return to normal before trading resumes.

Input	Purpose
Enable	Switches this detector on. Also requires the master Enable.
Timeframe	The chart timeframe the ATR is measured on.
ATR Period	Bars used in the ATR calculation.
Value Type (Flat / Points / Percent)	How the thresholds below are interpreted against the measured ATR.
Trip Threshold (0=Off)	Volatility at or above this trips the breaker. `0` = detector inactive.
Re-arm Threshold (0=Trip Threshold Used)	Volatility must fall back below this to re-arm. `0` = use the trip threshold, giving no hysteresis.

Market Anomaly > Correlated Asset Shock

Watches up to three independent **baskets** of symbols for an abnormally fast coordinated move — the signature of a currency-wide or market-wide event rather than one instrument having a moment.

The measure is a rate: percent moved per minute, not total percent. This matters. A total- percent threshold means something completely different at each horizon — 1% is extraordinary in five minutes and unremarkable in an hour — so the longest window would decide everything and the shorter ones would be decoration. A rate means the same thing at every horizon. These are all the same threshold expressed differently:

```
0.050 %/min   =   0.25% in 5 min   =   0.75% in 15 min   =   3% per hour
```

Rates are small numbers and easy to mis-enter by a factor of ten, so calibrate in whichever unit you think in and convert.

A basket is an AND. Every symbol in it must clear the rate *and* all must be moving the same way before that basket trips. That is precisely what separates a genuine correlated event from one symbol gapping on its own news. A basket containing a single symbol is therefore just a plain single-symbol shock detector — which is exactly how to watch an index or a metal alongside an FX basket.

The `-` prefix inverts a symbol's direction, and without it most baskets could never trip. Consider watching for a USD shock: a strong dollar means USDCHF *rising* but EURUSD *falling*. Written plainly those two permanently disagree and the basket never fires. Written as `USDCHF,-AUDUSD,-EURUSD` all three point the same way when the dollar moves. A `+` prefix is accepted and does nothing, purely so a mixed list reads evenly.

The horizons exist for redundancy, not to span event durations. Several closely-spaced lookbacks mean a fast move is caught whichever window happens to align with it, rather than being missed because it straddled a single window's boundary. Keep them close together — under a rate threshold a widely-spaced long horizon almost never fires, since the same burst averaged over four hours is a low rate. Note also that the longest horizon sets an implicit floor on re-arm: the move keeps registering until it has scrolled out of every window.

Individual baskets need no enable of their own — clear a basket's Symbols field, or leave its rate at `0`, and it is off.

Input	Purpose
Enable	Switches this detector on. Also requires the master Enable.
Lookback Horizons (Minutes; CSV)	The windows each rate is measured over, shared by all baskets. Several close together give redundancy.
Re-arm Level (Fraction of Trip)	Hysteresis, as a fraction of each basket's own trip rate. `1.0` = re-arm as soon as the rate falls back under the trip level. `0.33` = it must fall to a third of it first. Expressed as a fraction so it can never drift out of sync with a retuned threshold.
Basket 1: Symbols (CSV; Empty=Off; '-'=Invert)	The symbols that must all move together. Empty = this basket is off.
Basket 1: Trip Rate (% per Minute; 0=Off)	The rate every symbol in basket 1 must clear. `0` = off.
Basket 2: Symbols / **Basket 2: Trip Rate**	As basket 1, independently configured.
Basket 3: Symbols / **Basket 3: Trip Rate**	As basket 1, independently configured.

Results

Controls which closed trades the GUI's Results tab reports on.

The tab aggregates realised closed-trade history into Day, Week, Month and All-time totals. Unlike most of the EA, this reporting is **account-wide across every symbol** rather than limited to the chart — it is a performance read-out, not a trading function, so it reports on everything the magic numbers cover wherever those trades were taken.

The EA's own magic is always included. This setting only adds *further* magics to the report. That makes it useful for seeing several instances' combined performance from any one of their charts: list the other instances' magic numbers here and the tab totals them together.

Adding magics here changes reporting only. It never gives this instance any control over those trades — position management, drawdown limits and close operations all remain strictly limited to the EA's own magic.

Input	Purpose
Extra Magics (CSV; EA Magic Always Included)	Comma-separated magic numbers to include in the Results tab alongside this EA's own. Empty = this instance only.

Trade Comment

The comment attached to each order when it opens. It appears against the trade in MetaTrader's Trade and History tabs and in broker statements, which makes it the label your trades carry outside the EA.

It is written once, at open. Closes and modifications never change it, so a position keeps the comment it was born with for its whole life.

The preset formats are built from facts the EA already knows — magic number, direction, lot size — in the combinations below. `Custom` sends your own text verbatim instead.

Comment Type	Produces
`None`	no comment
`Magic`	`Magic 1001`
`Direction`	`BUY`
`Magic + Direction`	`Magic 1001, BUY`
`Direction + Lots`	`BUY, 0.25 lots`
`Magic + Direction + Lots`	`Magic 1001, BUY, 0.25 lots`
`Custom`	your text, exactly as entered

There is deliberately no "detailed" format carrying the signal source or entry context. That level of information belongs in the log, where it is available in full when Detail Level is raised, rather than squeezed into a short field the broker may truncate.

Input	Purpose
Comment Type	Which format to attach — see the table above.

Input	Purpose
Custom Comment (Used When Type = Custom)	Your own comment text. Read only when Comment Type is `Custom`. Brokers may truncate long comments.

Logging

How much the EA writes to MetaTrader's Experts log.

One verbosity dial, not a tree of toggles. The levels are **cumulative** — each includes everything below it — so raising the level never removes a line you were already getting.

Level	Adds
`Off (Safety Alerts Only)`	Nothing beyond the safety notices described below
`Trades (Opens, Closes, Rejections)`	Orders sent, with margin and the broker's return code, plus fills, opens and closes
`Detailed (Adds Signals and SL/TP Changes)`	Entry signals, stop and target change requests and their outcomes, and a per-tick state dump that stays quiet until something actually changes
`Diagnostic (Everything, Every Tick)`	The same dumps unthrottled, every tick, uncapped

`Detailed` is the useful everyday ceiling. `Diagnostic` is a genuine firehose intended for reproducing a specific problem, not for leaving switched on — it will fill the log quickly and slow the terminal down.

Mass closes always log, at every level including `Off`. A drawdown, anomaly or news flatten is safety-critical, so it is deliberately not something this setting can silence. If the EA ever closes everything, you will find out why in the log regardless of how it is configured.

For diagnosing a configuration that will not trade, `Detailed` is the level that shows entry signals and the reason a trade was rejected — the log counterpart to the Entry Gates tab.

Input	Purpose
Detail Level	`Off`, `Trades`, `Detailed`, or `Diagnostic` — cumulative, as above.

Notifications

Get notified on trades and important events, on your phone or at the terminal.

Notifications are a live-trading feature only. They do nothing in the Strategy Tester, during either a backtest or an optimisation.

What triggers a message

Three event types, each routable to each channel independently:

Event	Sent when
Position Opened	The **first** position opens on a side
Trading Halted or Resumed	A drawdown limit or anomaly breaker halts trading, the EA arms with invalid inputs, or trading resumes
Position Closed	A side goes **flat** — the whole cycle is finished

Entries and exits are reported per cycle, not per position. One message when a side opens and one when it goes flat — not one per position in a scaling structure, and not one per partial close. The closing message carries the side's net result and how many positions it took.

Every halt is followed by a resume message, so a halt notification never leaves the state ambiguous.

The invalid-inputs message is the most important one the EA sends. An EA that armed but cannot trade otherwise looks identical to an EA waiting for a signal.

The two channels

Channel	Goes to	Delivery
Push	The MetaQuotes mobile app	Queued and released at the platform's accepted rate
Alert	A popup on the machine running the EA	Immediate

MetaQuotes rate-limits push notifications to roughly two per second and discards anything beyond that. The EA therefore queues pushes and releases them at an accepted rate, so a burst during a halt arrives slightly delayed rather than being lost. A message that fails to send — a briefly offline terminal, for instance — is retried rather than discarded, and order is preserved.

Push requires a MetaQuotes ID set in the terminal, under *Tools* > *Options* > *Notifications*. This cannot be configured from the EA. If any push event is enabled while the terminal has no ID set, the EA reports it as an invalid input on the Status tab.

Message format

Every message opens with the product name and the chart symbol, followed by an optional instance identifier:

```
Aeterna EURUSD: BUY 0.25 opened @ 1.09341
Aeterna EURUSD #1001: HALTED — Daily limit hit. Positions closed. Cooldown active.
Aeterna EURUSD [Breakout]: SELL closed +42.18 USD (4 legs)
```

Identifier	Renders as	Use when
`None`	`Aeterna EURUSD:`	Only one instance is running
`Magic`	`Aeterna EURUSD #1001:`	Several instances, distinguished by magic number
`Strategy Name`	`Aeterna EURUSD [Breakout]:`	Several instances with meaningful names
`Account Number`	`Aeterna EURUSD (5182204):`	The same EA and magic run across several accounts — common in prop-firm use

Selecting `Strategy Name` while the Strategy Name field is empty is refused, and reported on the Status tab.

Push messages are capped at 255 characters by the platform. Messages are built to fit within that limit, so nothing is lost to truncation.

Input	Purpose
Instance Identifier	`None`, `Magic`, `Strategy Name`, or `Account Number` — appended after the symbol on every message.

Notifications > Push

Which events reach the MetaQuotes mobile app. Requires a MetaQuotes ID in the terminal's own options, as described above.

Input	Purpose
Position Opened (First Per Side)	Push when a side opens its first position.
Trading Halted or Resumed	Push on halts, the invalid-inputs warning, and resumes.
Position Closed (With Result)	Push when a side goes flat, with net result and leg count.

Notifications > Alert

Which events raise a popup on the machine running the EA. Independent of the push switches, so you can watch entries at the desk while only halts reach your phone.

Input	Purpose
Position Opened (First Per Side)	Popup when a side opens its first position.
Trading Halted or Resumed	Popup on halts, the invalid-inputs warning, and resumes.
Position Closed (With Result)	Popup when a side goes flat, with net result and leg count.

Backtest Statistics

Exports a backtest's performance as a **time series**, decomposed into fixed intervals, to a CSV file for analysis outside MetaTrader.

Why decompose a single run

The Strategy Tester reports a backtest as a set of aggregate figures covering the entire period. Those aggregates answer whether a configuration was profitable overall, but they cannot show **how that result was distributed across time** — and distribution is what separates a robust configuration from a fragile one.

Two configurations returning identical totals may have arrived there very differently. One may have produced modest, consistent gains across most intervals; the other may have been flat or negative throughout and rescued by a single exceptional stretch. The aggregate cannot tell them apart. Broken into intervals, they are obviously different — and the second is far more likely to have fitted a specific market episode rather than captured a durable edge.

Interval decomposition supports the analysis that aggregates preclude:

- **Consistency** — the dispersion of returns across intervals, not merely their sum. A high mean with high variance is a materially different proposition from the same mean with low variance.
- **Regime dependence** — whether results cluster in particular conditions. Persistent profitability across varied intervals is evidence of an edge; profitability confined to one stretch is evidence of fitting.
- **Concentration** — whether the total is carried by a small number of outlier intervals. A result driven by one interval is a result driven by one episode.
- **Deterioration** — whether the recent portion of a run resembles the earlier portion, which a single figure spanning both conceals entirely.

It also records measures at finer granularity than the Tester's summary: per-interval equity extremes with the times they occurred, drawdown measured two distinct ways, and profit attributed by both open and close date.

Behaviour and output

Backtesting only. The module is inactive in live trading and during optimisation. It stands down entirely for optimisation passes — those run in parallel across separate processes that would race to write the same file, and the optimiser's own results table is the appropriate artifact there. This module exists to decompose a **single** run.

The export is written to `Terminal\Common\Files\` as `Aeterna_Strategy_Engine_Backtest_Stats.csv`, with one row per interval. It is written when a run completes, and also when a run is cancelled part-way, so an interrupted test still yields its data.

Input	Purpose
Enable	Switches the module on. Has no effect outside a single (non-optimisation) backtest.

Input	Purpose
Interval	The aggregation interval: `Daily`, `Weekly (Monday start)`, `Monthly`, or `Entire Backtest`. Shorter intervals give more observations and finer resolution; longer ones give less noise per row. `Entire Backtest` produces a single row and forgoes decomposition.
Export Latest First (Else Oldest First)	Row ordering — reverse-chronological or chronological.

Choosing an interval is a sample-size decision. Shorter intervals yield more observations, which supports firmer conclusions about consistency, but each row then covers fewer trades and is correspondingly noisier. A strategy taking a handful of trades per month produces largely empty rows on a daily interval.

Backtest Statistics > Performance

Trade-outcome measures per interval.

Realized profit is available under two attribution conventions, and they answer different questions. *Close Date* assigns a position's profit to the interval it closed in — the standard convention, and the one that reconciles with account statements. *Open Date* assigns it to the interval it was opened in, which attributes the result to the decision that produced it rather than to whenever it happened to conclude. For strategies holding positions across interval boundaries the two series differ, and comparing them isolates how much of an interval's result originated in decisions taken earlier.

Input	Purpose
Realized Profit (Close Date)	Net profit of positions **closed** in the interval.
Realized Profit (Open Date)	Net profit of positions **opened** in the interval, regardless of when they closed.
Win Rate	Proportion of closed positions that were profitable.
Profit Factor	Gross profit divided by gross loss. Above 1.0 is profitable; the ratio expresses how much was won per unit lost.
Largest Win	Largest single winning position.
Largest Loss	Largest single losing position.
Average Win	Mean profit across winning positions.
Average Loss	Mean loss across losing positions.

Win rate and profit factor are complementary, not interchangeable. A low win rate with a high profit factor describes a strategy taking many small losses against occasional large gains; the inverse describes one taking frequent small gains against occasional large losses. Average win and average loss give the magnitudes underlying both, and together they characterise the return distribution far better than either ratio alone.

Backtest Statistics > Equity

The equity path within each interval, rather than only its endpoints.

Input	Purpose
Equity Start	Equity at the interval's open.
Equity End	Equity at the interval's close.
Equity High	Peak equity reached during the interval.
Equity High Time	When that peak occurred.
Equity Low	Trough equity reached during the interval.
Equity Low Time	When that trough occurred.
Equity Change	Net movement from start to end.

Start and end alone understate the path taken. An interval finishing where it began may have been uneventful, or may have travelled a long way in both directions first. The high and low capture that excursion, and their timestamps place it — which is what allows an equity trough to be matched against a known market event.

Backtest Statistics > Drawdown

Drawdown measured two ways per interval. The distinction matters and is frequently conflated.

Measure	Measured from	Captures
Highest Balance to Lowest Equity	the interval's peak **balance** (realised)	The full excursion including open floating loss — the deepest point relative to banked capital
Highest Equity to Lowest Equity	the interval's peak **equity** (including floating)	The peak-to-trough decline of the equity curve itself

The first is generally the larger figure and the one most prop-firm rules are written against, since it measures against realised capital. The second describes the equity curve's own volatility. Each is available in both account currency and as a percentage.

Input	Purpose
Max DD (Highest Balance to Lowest Equity)	Largest balance-to-equity decline, in account currency.
Max DD % (Highest Balance to Lowest Equity)	The same, as a percentage.
Max DD (Highest Equity to Lowest Equity)	Largest equity-to-equity decline, in account currency.
Max DD % (Highest Equity to Lowest Equity)	The same, as a percentage.

Backtest Statistics > Trades

Position counts per interval — the denominators for the performance measures above.

Input	Purpose
Positions Opened in Period	Positions opened during the interval, whenever they closed.
Positions Closed in Period	Positions closed during the interval, whenever they opened.
Positions Opened and Closed in Period	Positions that both opened and closed within the interval.

These counts qualify every other figure in the row. A profit factor computed from three closed positions carries far less weight than the same figure computed from ninety, and without the count the two are indistinguishable in the export. The third measure additionally indicates holding period relative to the interval: where it approaches the closed count, positions are resolving within the interval; where it is much lower, positions are spanning boundaries and the two attribution conventions above will diverge.

Reference

Worked configurations

Adapting to prop firm drawdown rules

Troubleshooting

Glossary