Kestutis Balciunas

NINJATRADER 8 INDICATOR · NINJASCRIPT, C#

User Guide

Market structure with BOS and CHoCH, order blocks that extend until mitigated, retest signals on closed bars with an entry, a stop and two targets, four alerts, and series a strategy can read.

PRODUCT	SMC Algo Warrior for NinjaTrader 8
VERSION	1.0.0
DATE	November 2026
NINJATRADER	8.1.x Desktop, tested on 8.1.9.0
PACKAGE	`SmcAlgoWarrior.zip`, compiled NinjaScript Add-On
SHA-256	`5606ef3e4b7514548a3c400706c765be6b1eee45a712c2b701b6515ffdd3f891`
AUTHOR	Kestutis Balciunas, https://kestutisbalciunas.com/
PRODUCT PAGE	https://kestutisbalciunas.com/ninjatrader/smc-algo-warrior/
SUPPORT	https://kestutisbalciunas.com/contact/

Trading involves substantial risk. Futures, foreign currency and options trading contains substantial risk and is not for every investor. Past, simulated and backtested performance is not necessarily indicative of future results.

Contents

01 What it is

02 Installation

03 Quick start

04 Calculate mode and bar timing

05 Outputs for strategies and the Market Analyzer

06 Parameters

07 Sessions, time and volume

08 Compatibility

09 Troubleshooting

10 How this edition was verified

11 Licensing

12 Support

13 Risk disclosure

01

What it is

Chart indicator for NinjaTrader 8 Desktop (NinjaScript, C#) for market structure and order-block retests. It finds swing highs and lows, marks each close through a swing as a break of structure (BOS) or a change of character (CHoCH), builds an order block on the break, and prints a buy or sell arrow when a closed bar retests an active block, with an entry line, a stop and two targets. It has four alert conditions and five series a strategy can read.

It only draws and alerts: it never places, modifies or closes orders. A webhook is not part of this edition: the indicator makes no network call of its own.

02

Installation

Requirements

- NinjaTrader 8.1.x Desktop for Windows (tested on 8.1.9.0). Not for NinjaTrader 8.0 or NinjaTrader Web.
- Any data connection that delivers bars for the chart. The calculation does not read volume. No Order Flow+ features, no depth, no tick history.

Import the Add-On package

1. Control Center > Tools > Import > NinjaScript Add-On.
2. Select `SmcAlgoWarrior.zip` and confirm. NinjaTrader compiles the assembly and reports that the import is complete.
3. No restart is needed. If a chart already had an older version, remove and re-add the indicator (a compile does not reload running instances).

Verify installation

1. Open a chart with at least a few hundred bars (any instrument, a 1 to 60 minute interval is a good first look), press Ctrl+I.
2. In the Available list find **SMC Algo Warrior**, click add, click OK.
3. The chart shows structure lines labelled BOS and CHoCH, translucent green and red order-block boxes, and, where a closed bar retested a block, a buy arrow under the bar or a sell arrow over it. The five newest signals also carry an entry line, an SL line, two TP lines and a price tag on each. Mitigated blocks are removed by default, so a chart can show few boxes at a time.

Uninstall / update

- Remove: Control Center > Tools > Remove NinjaScript Assembly > SMC Algo Warrior.
- Update: import the new zip over the installed version (NinjaTrader replaces the assembly), then remove and re-add the indicator on every chart that uses it, or reload the workspace.

03

Quick start

First chart

1. Add **SMC Algo Warrior** to a chart (Ctrl+I). Leave every setting at its default.
2. Load enough history. The swing state, the trend and the active order blocks on screen are built from the bars before them, and the indicator does not extend the chart's data: the Data Series setting **Days to load** decides. Aim for at least 5,000 bars where your feed allows it.
3. Everything is decided on closed bars. Nothing is calculated or drawn on the forming bar.

Reading the chart

- **Structure lines**: a close above the last unbroken swing high is an upward break, a close below the last unbroken swing low a downward break. A break against the structure trend is a **CHoCH** (solid line); any other break, the very first one included, is a **BOS** (dotted line). Teal for an upward break, orange for a downward one. The line runs from the swing to the break bar, and the label sits at its middle: above it for an upward break, below it for a downward one.
- **Order blocks**: each break creates a box. For an upward break it is the candle with the lowest low since the last swing low (the last 20 bars when there is no swing low yet, never more than 100 bars back); for a downward break, the candle with the highest high since the last swing high. Green is bullish, red is bearish. The box grows to the right until a close goes through it, below a bullish block or above a bearish one, and is then removed. At most four blocks per side stay active.
- **Arrows**: a green arrow under the bar is a buy, a magenta arrow over the bar is a sell.
- **Levels**: a signal draws a dotted grey entry line, a red SL line, a green TP1 line and a cyan TP2 line, 20 bars long, with a price tag at the right end of each. The newest five sets stay. A tag that would fall past the right edge is pulled inside the chart and its line ends where the tag starts.
- **The levels are a plan, not a tracker**: they are drawn when the arrow prints and stay as they are. The indicator does not follow them, mark a target or the stop as reached, or count outcomes.

Signal rules in one paragraph

A **buy** prints on an active bullish block when the bar is later than the bar that created the block, its low reaches the block's top or below, it closes above its open, it did not close below the block, the block has not given a signal before, no buy has printed on this bar yet, and, with **only trade in direction of structure** on (the default), the structure trend is up. A **sell** is the mirror image on a bearish block. Each block gives at most one signal, and each bar at most one buy and one sell. The entry is the signal bar's close. The SL sits

sl buffer (x atr) ATRs beyond the block's far edge, or beyond the bar's own extreme if that is further. TP1 and TP2 are the entry plus or minus **tp1 risk:reward** and **tp2 risk:reward** times the distance from the entry to the SL. Until the ATR has **atr length** bars, a signal prints without levels.

Settings to try first

SETTING	WHAT IT CHANGES
swing length	Bars on each side of a swing. Larger values find fewer swings, so fewer breaks and fewer blocks.
create ob only on choch (reversal)	Creates a block only on a CHoCH, never on a BOS.
only trade in direction of structure	Off allows a retest against the structure trend to print.
remove ob when mitigated	Off keeps a mitigated block in grey, no longer extending. It can no longer give a signal.
show bos / show choch / show structure lines + labels	Hide the structure drawing only. The breaks, the trend, the blocks and the signals are unchanged.
show order blocks	Off creates no blocks. Every signal is a retest of a block, so no signals or levels appear either.

Alerts

Four events, each behind its own switch in **07 Alerts**, all on by default. They are evaluated once per closed bar and only in realtime, never during the history pass, and never for a bar that closed more than two bar periods plus five seconds ago.

SWITCH	FIRES WHEN	MESSAGE
Alert: buy: bullish ob retest	a buy signal prints	`BUY (bullish OB retest) on MNQ 5 @ 29307.25`
Alert: sell: bearish ob retest	a sell signal prints	`SELL (bearish OB retest) on MNQ 5 @ 29307.25`
Alert: bullish choch	an upward change of character	`Bullish CHoCH on MNQ 5 @ 29307.25`
Alert: bearish choch	a downward change of character	`Bearish CHoCH on MNQ 5 @ 29307.25`

The message carries the instrument, the chart interval (`5`, `60`, `1D`, `30S`, `500T`) and the close of the bar at the instrument's tick size. A BOS has no alert. The drawing switches do not silence an alert: **show entry arrows** off still alerts the signal, and **show structure lines + labels** off still alerts a CHoCH.

CHANNEL	DEFAULT	WHAT ARRIVES
Alerts Log	always	one row per alert
Alert Log Line	on	the same line in the NinjaScript Output window
Alert Sound	off	the file named in **Alert Sound File**: a name inside NinjaTrader's sounds folder or a full path; blank plays `Alert2.wav`
Alert E-mail	off	an e-mail to **E-mail To**, through NinjaTrader's Mail share service (Tools > Options > Share)

The row and the Output line read `buy: bullish ob retest | BUY (bullish OB retest) on MNQ 5 @ 29307.25`. An e-mail has the subject `SMC Algo Warrior | buy: bullish ob retest` and the message as its body.

Test Alerts on Start (off) sends one test through these channels when the indicator goes live, for example `test | SMC Algo Warrior test on MNQ 5 @ 29307.25`. Applying the Indicators dialog recreates the instance, so every OK sends one more test while it is on.

A webhook is not part of this edition: the indicator makes no network call of its own.

The final bar of a session

A signal on a session's final bar appears after the reopen. NinjaTrader closes a bar when the next bar's first tick arrives, so on a closed market that last bar is calculated and drawn only at the first tick after the reopen. Its alert is raised only if the reopen comes within two bar periods plus five seconds of the bar's close; otherwise the arrow prints with no alert.

Using the values in a strategy or the Market Analyzer

Five hidden series are exposed: `BuySignal`, `SellSignal`, `BullishChoch` and `BearishChoch` (1 on the closed bar of the event, 0 otherwise) and `StructureTrend` (1, -1 or 0). Example: `if (SmcAlgoWarrior(...).BuySignal[0] > 0)`. With `OnEachTick` or `OnPriceChange` the bar that just closed is `[1]`.

What it does not do

It places no orders and sends nothing over a network of its own. It draws what the closed bars show; it does not forecast where price goes next.

04

Calculate mode and bar timing

- Default `Calculate.OnBarClose`: one engine commit per closed bar.
- `OnEachTick` / `OnPriceChange`: the closed bar is committed once, at the first tick of the next bar. The forming bar is not calculated in any mode, so these modes draw the same objects on the same bars and do not make a signal earlier.
- Nothing is drawn on a forming bar. An arrow or a break does not change once drawn. An order block grows one bar to the right on every bar until it is mitigated or replaced. A swing is confirmed **swing length** bars after its own bar: that is a delay, and the break it allows can only happen after the confirmation.
- **A signal on a session's final bar appears after the reopen.** NinjaTrader closes a bar when the next bar's first tick arrives, so on a closed market the session's last bar stays uncommitted until the first tick after the reopen, and only then is it calculated and drawn. Its alert is raised only if the reopen comes within two bar periods plus five seconds of that bar's close; otherwise the arrow prints with no alert.
- Historical pass: one call per closed bar. A still-forming last bar is not committed. The engine slot is the NinjaTrader bar index and the host walks it from the `Bars` series, so the historical to realtime transition can neither double-feed nor skip a bar; a skipped range is refilled from the series.
- With `OnEachTick` / `OnPriceChange`, Tick Replay takes the same path as per-tick realtime.
- The indicator keeps calculating while its chart window is minimized (`IsSuspendedWhileInactive = false`), and an alert is raised only for a bar that closed within the last two bar periods plus five seconds, so a catch-up after a suspension or a reconnect does not replay old alerts. On tick, volume, range and other non-time bars that limit is at least five minutes.
- A bar's OPEN time is its NinjaTrader close time minus the bar period (on non-time bars, the smallest spacing of the first 300 bars). Only the alert's once-per-bar key and the CSV dump read it.
- The price axis follows the bars, not the drawings (`IsAutoScale = false`), so a far stop or target can lie outside the panel.

05

Outputs for strategies and the Market Analyzer

PLOT / SERIES	VALUES
`BuySignal` (plot 0, hidden)	1 on the closed bar of a buy signal, 0 otherwise
`SellSignal` (plot 1, hidden)	1 on the closed bar of a sell signal, 0 otherwise
`BullishChoch` (plot 2, hidden)	1 on the closed bar of an upward change of character, 0 otherwise
`BearishChoch` (plot 3, hidden)	1 on the closed bar of a downward change of character, 0 otherwise

PLOT / SERIES	VALUES
`StructureTrend` (plot 4, hidden)	the structure trend after the bar: 1 after an upward break, -1 after a downward one, 0 before the first break

The series are not drawn. They follow the signals and not the drawing switches, so **show entry arrows** off leaves `BuySignal` and `SellSignal` unchanged.

With `Calculate.OnBarClose`, `[0]` is the bar that just closed. With `OnEachTick` / `OnPriceChange`, the bar that just closed is `[1]`; on the forming bar `[0]` reads 0 for the four events, and `StructureTrend[0]` holds the trend as of the last close.

Example in a strategy: `var saw = SmcAlgoWarrior(...); if (saw.BuySignal[0] > 0) ...` (the accessor is generated by NinjaTrader on compile; every input except the 13 colours is one of its arguments).

06

Parameters

The 32 inputs in groups 01 to 05 keep the titles, order, defaults and ranges of the other editions (the group names carry a number prefix because NinjaTrader sorts groups alphabetically). Groups 06 Display, 07 Alerts and 08 NinjaTrader hold the 14 platform settings: 46 parameters in total.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
01 market structure (bos / choch)	swing length	int	5	from 1	`SwingLen`	Bars on each side a swing high or low needs. A swing is confirmed this many bars after its own bar, so larger values find fewer, more significant swings.
01 market structure (bos / choch)	show structure lines + labels	bool	true		`ShowStructure`	Draws the break lines and their BOS / CHoCH labels. Drawing only: off, the breaks, the structure trend, the order blocks and the signals are unchanged.
01 market structure (bos / choch)	show bos	bool	true		`ShowBOS`	Draws breaks that continue the structure: a dotted line from the swing to the break bar, labelled BOS.
01 market structure (bos / choch)	show choch	bool	true		`ShowCHoCH`	Draws breaks that reverse the structure: a solid line, labelled CHoCH.
02 order blocks	show order blocks	bool	true		`ShowOB`	Creates an order block on each break. Off creates none, and because every signal is a retest of an order block, no signals, levels or signal alerts appear either.
02 order blocks	create ob only on choch (reversal)	bool	false		`ChochOnly`	On: an order block is created only on a CHoCH, never on a BOS.
02 order blocks	only trade in direction of structure	bool	true		`TrendFilter`	On: a buy retest needs a bullish structure (the last break was upward), a sell retest a bearish one.
02 order blocks	max active ob per side	int	4	1 to 20	`MaxOB`	How many order blocks of each side stay active. When a new one exceeds the number, the oldest is removed.
02 order blocks	remove ob when mitigated	bool	true		`RemoveMit`	On: an order block is deleted once a bar closes through it (below a bullish block, above a bearish one). Off: it stays in the mitigated colour and stops extending.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
03 signals	show entry arrows	bool	true		`ShowArrows`	Draws the buy arrow under the bar and the sell arrow over it. Drawing only: the signals, their levels, the alerts and the series are unchanged.
04 sl / tp	show sl/tp with arrows	bool	true		`ShowSLTP`	Draws the entry, SL, TP1 and TP2 lines with their price tags for every signal.
04 sl / tp	atr length	int	14	from 1	`AtrLen`	Length of the ATR (Wilder's average true range) behind the SL buffer. Until it has enough bars, a signal prints without levels.
04 sl / tp	sl buffer (x atr)	double	0.2	from 0.0	`SlBuffer`	How far beyond the order block's far edge, or the signal bar's own extreme if that is further, the SL sits, in ATRs.
04 sl / tp	tp1 risk:reward	double	1.0	from 0.1	`Tp1RR`	TP1 distance from the entry, as a multiple of the entry-to-SL distance.
04 sl / tp	tp2 risk:reward	double	2.0	from 0.1	`Tp2RR`	TP2 distance from the entry, as a multiple of the entry-to-SL distance.
04 sl / tp	show tp2	bool	true		`ShowTP2`	Draws the TP2 line and tag.
04 sl / tp	sl/tp line length (bars)	int	20	from 3	`TradeLen`	How many bars to the right of the signal bar the level lines run; each price tag sits at its line's right end.
04 sl / tp	max trades kept on chart	int	5	1 to 50	`MaxTrades`	How many signals keep their level lines and tags. When a new set exceeds the number, the oldest set is removed.
04 sl / tp	show price in labels	bool	true		`ShowPrices`	Writes the price on each tag; off shows the names only.
05 colors	bullish ob fill	colour	#4CAF50		`BullOBBrush`	Fill of a bullish order block (default alpha 38 of 255).
05 colors	bullish ob border	colour	#4CAF50		`BullOBBorderBrush`	Border of a bullish order block (default alpha 128 of 255).
05 colors	bearish ob fill	colour	#F23645		`BearOBBrush`	Fill of a bearish order block (default alpha 38 of 255).
05 colors	bearish ob border	colour	#F23645		`BearOBBorderBrush`	Border of a bearish order block (default alpha 128 of 255).
05 colors	mitigated ob (if kept)	colour	#787B86		`MitBrush`	Fill and border of a mitigated order block when remove ob when mitigated is off (default alpha 31 of 255).
05 colors	bullish bos/choch	colour	#089981		`BullStructBrush`	Line and label of an upward break (BOS or CHoCH).
05 colors	bearish bos/choch	colour	#FF9800		`BearStructBrush`	Line and label of a downward break (BOS or CHoCH).
05 colors	buy arrow	colour	#00E676		`BullArrowBrush`	The buy arrow.
05 colors	sell arrow	colour	#E040FB		`BearArrowBrush`	The sell arrow.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
05 colors	entry line	colour	#787B86		`EntryBrush`	The dotted entry line and its tag.
05 colors	sl line	colour	#F23645		`SlBrush`	The SL line and its tag.
05 colors	tp1 line	colour	#4CAF50		`Tp1Brush`	The TP1 line and its tag.
05 colors	tp2 line	colour	#00BCD4		`Tp2Brush`	The TP2 line and its tag.
06 Display	Keep Chart Object Limits	bool	true		`KeepChartObjectLimits`	On: only the newest 500 lines, 500 labels and 200 boxes stay on the chart and older ones are retired. Off: every structure line and label since the first bar stays.
06 Display	Clamp Right-Edge Labels	bool	true		`ClampRightEdgeLabels`	Keeps the entry, SL and TP price tags inside the chart when their bar lies past the right edge; the line ends where its tag starts.
07 Alerts	Alert: buy: bullish ob retest	bool	true		`AlertBuy`	Delivers the buy signal (a bullish order-block retest). Every alert appears in the Alerts Log window.
07 Alerts	Alert: sell: bearish ob retest	bool	true		`AlertSell`	Delivers the sell signal (a bearish order-block retest).
07 Alerts	Alert: bullish choch	bool	true		`AlertBullChoch`	Delivers the upward change of character.
07 Alerts	Alert: bearish choch	bool	true		`AlertBearChoch`	Delivers the downward change of character.
07 Alerts	Alert Log Line	bool	true		`AlertLogLine`	Writes each alert to the NinjaScript Output window.
07 Alerts	Alert Sound	bool	false		`AlertSound`	Plays a sound with each alert.
07 Alerts	Alert Sound File	string	"Alert2.wav"		`AlertSoundFile`	A file name inside the NinjaTrader sounds folder, for example Alert2.wav, or the full path to a .wav file. Blank plays Alert2.wav.
07 Alerts	Alert E-mail	bool	false		`AlertEmail`	Sends each alert by e-mail through the Mail share service configured in NinjaTrader (Tools > Options > Share).
07 Alerts	E-mail To	string	""		`EmailTo`	Recipient address for alert e-mails. Blank sends none.
07 Alerts	Test Alerts on Start	bool	false		`AlertTestOnStart`	Sends one test alert through every channel switched on when the indicator goes live, so the sound and the e-mail can be checked without waiting for a signal.
08 NinjaTrader	Debug Mode	bool	false		`DebugMode`	Writes diagnostics to saw_nt_debug.log in the NinjaTrader 8 folder.
08 NinjaTrader	Dump Bars + Outputs CSV	bool	false		`DebugDumpCsv`	Writes every committed bar with the engine outputs to saw_dump_<instrument>_<period>.csv (parity checks).

46 parameters.

07

Sessions, time and volume

Sessions

The indicator has no session logic of its own: every bar the chart delivers is fed to the engine in order, so the chart's Trading Hours template decides which bars exist. With an ETH template (for example `CME US Index Futures ETH`) the overnight bars take part in the swings, the breaks, the order blocks and the ATR; with an RTH template they do not. Both are valid, and the structure differs, as it does for any indicator. A structure line or an order block across a session gap simply spans the bars on both sides.

The final bar of a session

A signal on a session's final bar appears after the reopen. NinjaTrader closes a bar when the next bar's first tick arrives, and the indicator calculates a bar only once it has closed, so on a closed market the last bar stays uncommitted until the first tick after the reopen. It is then calculated and drawn. Its alert is raised only if the reopen comes within two bar periods plus five seconds of that bar's close; otherwise the arrow prints with no alert. The same applies to any later close on a quiet instrument: the bar is committed when the next tick starts a new bar.

History

The swing state, the trend and the active order blocks on screen are built from the bars before them, starting from the first bar the chart loaded. The indicator does not extend the chart's data, so the Data Series setting **Days to load** decides how much history is behind them. Two charts that loaded different amounts of history can differ for a while from where the shorter one starts. With **Keep Chart Object Limits** on, the newest 500 lines, 500 labels and 200 boxes stay; the counts can run up to five over before the oldest are retired.

Time stamps

NinjaTrader stamps a bar with its CLOSE time in the platform time zone. The bar's open time is reconstructed as close time minus the bar period; on non-time bars the smallest spacing of the first 300 bars stands in for the period. Only the alert's once-per-bar key and the CSV dump read it. Everything drawn is positioned by bar index. Slots past the last bar, where the level lines and their tags usually end, are placed at the average bar spacing of the view.

Volume

The calculation does not read volume: swings, breaks, order blocks, signals and levels come from open, high, low and close alone. The CSV dump writes the bar's volume for bar identity only.

Contract rollover

A continuous contract (merge policy `MergeBackAdjusted` or `MergeNonBackAdjusted` in the Data Series dialog) shows a price jump at the roll unless back-adjusted. Swings, breaks, order blocks and the ATR behind the stop are built from prices, so a non-back-adjusted roll can produce a false break and an oversized ATR for a while; with back adjustment there is no jump. State the merge policy when comparing values across machines.

08

Compatibility

ITEM	STATUS
NinjaTrader version	8.1.x Desktop (tested 8.1.9.0). Not 8.0, not NinjaTrader Web or mobile.
Product type	Indicator, price panel overlay, single data series.
Instruments	Any instrument with bars: futures, micro futures, forex, stocks. Verified on MNQ (CME) 1-minute.
Bar types	Time-based (second, minute, day, week, month, year) fully. Tick, volume, range, Renko and other non-time bars run; on them the alert's age limit is at least five minutes.
Calculate modes	OnBarClose (default), OnEachTick, OnPriceChange. The forming bar is not calculated in any mode: a bar is committed once, when it has closed. With OnEachTick or OnPriceChange, Tick Replay takes the per-tick path.
Session close	A signal on a session's final bar appears after the reopen, at the next bar's first tick. See the chapter on sessions, time and volume.
Multi-instance	Yes. Every instance paints its own surface, and every alert id carries the instrument, the period and an instance hash.
Market Analyzer	Loads with no drawing; the `BuySignal`, `SellSignal`, `BullishChoch`, `BearishChoch` and `StructureTrend` series are available as columns. Not advertised as a scanner.
Strategies	The five series can be referenced from a strategy. Every input except the colours is a `[NinjaScriptProperty]`.
Data requirements	Bars only. The calculation does not read volume, so a feed without volume changes nothing.
Rendering	SharpDX in `OnRender` with DPI scaling and full resource cleanup on render-target change and termination. No draw objects.
Dark and light charts	The colours are inputs. There is no automatic switch between chart backgrounds.
Templates and workspaces	All settings serialize (colours through the standard brush serializer).
Minimized chart	Keeps calculating and alerting while its chart window is minimized (`IsSuspendedWhileInactive` off). A bar that closed more than two bar periods plus five seconds ago is not alerted, so a reconnect or a catch-up does not produce a burst of stale alerts.
Webhook	Not part of this edition. The indicator makes no network call of its own; e-mail goes through NinjaTrader's Mail share service.

ITEM	STATUS
Order handling	None. The indicator never touches orders, positions or accounts.

09

Troubleshooting

SYMPTOM	CAUSE AND FIX
Nothing on the chart	Check that the licence is active for your NinjaTrader login and that the indicator is in the Configured list (Ctrl+I). It paints in the price panel: with the indicator in its own panel it is not visible.
No arrows	A signal needs an active order block, a retest and a close in the right direction. **show order blocks** off means no signals at all. **only trade in direction of structure** on blocks a retest against the trend. **show entry arrows** off hides the arrows only.
An arrow with no levels	**show sl/tp with arrows** is off, or the ATR did not yet have **atr length** bars when the signal printed. Only the newest **max trades kept on chart** sets stay, so older arrows have no levels.
No signal on the last bar	The bar is still forming, or the session has ended. A signal on a session's final bar appears after the reopen, at the next bar's first tick.
OnEachTick shows nothing earlier	By design. The forming bar is not calculated in any Calculate mode.
An order block disappeared	A close went through it (below a bullish block, above a bearish one), or a new block pushed out the oldest beyond **max active ob per side**. **remove ob when mitigated** off keeps mitigated blocks in grey.
The oldest structure lines are gone	**Keep Chart Object Limits** on keeps the newest 500 lines, 500 labels and 200 boxes. Off keeps every structure line and label since the first bar.
A price tag is not at its line's natural end	**Clamp Right-Edge Labels** on pulls a tag inside the chart when it would fall past the right edge, and the line ends where the tag starts. The price on the tag is exact. Off, tags past the edge need room: chart Properties > Right side margin.
No tags after scrolling back	A tag is pulled in only while its trade's own bar is in view, so later trades stay out of a scrolled-back chart.
A level lies outside the panel	The price axis follows the bars, not the drawings. Drag the price axis to see it.
Alerts do not fire	Alerts fire only in realtime, only for bars that close after the chart loaded, and only with the per-event switch on. A BOS has no alert. A bar that closed more than two bar periods plus five seconds ago is not alerted, and that includes a session's final bar after a long break.
Sound not played	**Alert Sound** must be on. **Alert Sound File** is a name inside `<NinjaTrader 8 install>\sounds\` or a full path to a .wav file.
No e-mail	**Alert E-mail** must be on with **E-mail To** filled in, and the Mail share service configured under Tools > Options > Share.
Looking for a webhook	A webhook is not part of this edition. The indicator makes no network call of its own.
One more test alert after every change	**Test Alerts on Start** is on, and applying the dialog recreates the instance. Switch it off once the channels are checked.

SYMPTOM	CAUSE AND FIX
Values differ from another platform on the same symbol	Bars differ by feed and session template; compare on identical bars (see the chapter on how the port was verified).
The picture differs between two charts	The structure on screen is built from the bars before it. Compare charts with the same **Days to load** and Trading Hours template.
Text is tiny or huge	The label size is fixed and follows the Windows display scaling.
An error line names SMC Algo Warrior in the Log tab	Set **08 NinjaTrader > Debug Mode** on, reproduce it, and send `Documents\NinjaTrader 8\saw_nt_debug.log` with the chart interval, instrument and NinjaTrader version.
Everything is drawn twice	Two instances are configured (Ctrl+I, Configured list). Remove one.

10

How this edition was verified

The verification has two stages, which measure different things and are reported separately. Stage 1 ran on forex and gold bars, stage 2 on NinjaTrader futures bars, so the two figures are never added together.

Stage 1: the engine

The platform-free engine that ships here is the same code as the cTrader edition's engine apart from its namespace and using lines. That engine was replayed bar by bar there on EURUSD M5, M15 and H1 and XAUUSD M5, with 22 input sweeps on EURUSD M5 for the values and 15 for the drawn objects. Values and flags were compared exactly, drawn objects field by field.

GATE	RUNS	COMPARED	MISMATCHES
every calculated series, every bar	26	18,010,146 cells	0
plots, their colours and the alert conditions	26	4,366,096 cells	0
drawn objects (boxes, lines, labels, arrows)	19	165,410 fields	0

Stage 2: NinjaTrader host vs the engine on identical NinjaTrader bars (2 November 2026)

With **Dump Bars + Outputs CSV** on, the indicator writes every committed bar (time, open, high, low, close, volume) together with the engine outputs as the host saw them: the swing and ATR series, the swing state and the structure trend, the signal and CHoCH flags, both order-block lists, the newest trade's levels, the drawn-object counts, the arrows with their colours and the four alert conditions. The same bars are replayed through a build of the identical engine outside NinjaTrader, with no host in between, and every column is compared.

RUN	BARS	SERIES	CELLS	MISMATCHES
MNQ DEC26 1 Minute, historical load, NinjaTrader 8.1.9.0, default inputs	4,595	41	188,395	0

Bar times matched on every row, and the replay reads its prices from the same file. On this run the engine produced 31 buy and 32 sell signals and 57 bullish and 57 bearish CHoCH events, on the same bars on both sides.

Live run, 2 Nov 2026: MNQ 1 Minute with Calculate set to OnEachTick, across the switch from history to live bars, with about 20 bars closing live: 4,169 bars, 41 series, 170,929 cells, 0 mismatches, 0 bar-identity mismatches.

What stage 2 proves: on a historical load and across the switch to live bars the host feeds the engine one closed bar per chart bar, in order, and the engine's outputs inside NinjaTrader equal a replay outside it. What it does not cover: how the surface paints is checked on a chart, not by this diff; and none of this says anything about future performance.

11

Licensing

License model

The product is distributed as a compiled NinjaScript Add-On and licensed through NinjaTrader's user-based Vendor Licensing (NinjaTrader 8.1.5.0 or later). A licence is tied to the customer's NinjaTrader account login, not to a machine: the same login works on every installation of NinjaTrader Desktop the customer uses, and no Machine ID has to be sent to the vendor. Licences are created in the NinjaTrader Ecosystem Vendor Dashboard against the customer's NinjaTrader account email address, with a licence type and an expiry date.

The indicator carries the `VendorLicense(productId)` call in its constructor, with the Product ID of its add-on in the Vendor Dashboard. NinjaTrader verifies the licence whenever the indicator is used.

No free trial

The product is sold as a paid licence only; no free trial is offered. A licence is issued to the buyer's NinjaTrader account email after payment, normally within a few hours.

Activation and transfer

Nothing to activate: log in to NinjaTrader with the licensed account. A new PC needs no transfer. NinjaTrader allows two simultaneous market-data sessions per account, which is what prevents a licence from being shared.

Behavior without a valid licence

NinjaTrader shows its own licence notice (invalid or expired) and stops calling the indicator's calculation and plot methods: the indicator draws nothing and raises no alerts, while its settings still load. It never places orders (it has no order code) and exposes no proprietary data. A licensing-server outage is handled by NinjaTrader's verification (it retries against a secondary server) and must never crash the chart.

Licensing is a practical access-control layer, not a guarantee against copying.

12

Support

Questions go through the contact page at https://kestutisbalciunas.com/contact/ with the NinjaTrader version, the instrument and the chart interval. For an error, switch **08 NinjaTrader > Debug Mode** on, reproduce it once and attach `saw_nt_debug.log` from the `Documents/NinjaTrader 8` folder; switch Debug Mode off afterwards.

The product page, with the documentation and the change log, is https://kestutisbalciunas.com/ninjatrader/smc-algo-warrior/. Updates and refunds follow the terms on that page.

13

Risk disclosure

Trading involves substantial risk. Past, simulated, and backtested performance is not necessarily indicative of future results.

Futures, foreign currency and options trading contains substantial risk and is not for every investor. An investor could potentially lose all or more than the initial investment. Risk capital is money that can be lost without jeopardizing one's financial security or lifestyle. Only risk capital should be used for trading and only those with sufficient risk capital should consider trading.

SMC Algo Warrior is a technical analysis tool. It draws market structure, order blocks, signals and trade levels computed from closed bars; it does not place, modify or close orders and it gives no trade advice. The entry, stop and target lines are a plan drawn when a signal prints: the indicator does not follow them or report whether they are reached. Nothing in the software or its documentation is a promise of returns, win rates or funded-account success.

NinjaTrader is a registered trademark of NinjaTrader Group, LLC. No NinjaTrader company has any affiliation with the owner, developer, or provider of the products or services described herein, or any interest, ownership or otherwise, in any such product or service, or endorses, recommends or approves any such product or service.