Kestutis Balciunas

NINJATRADER 8 INDICATOR · NINJASCRIPT, C#

User Guide

Structure breaks (CHoCH and BOS) filtered on seven checks and graded A to C, an adaptive trend ribbon, a trade plan with a stop and three targets, the OTE zone of the current leg, a six-timeframe dashboard and alerts.

PRODUCT	Supreme Power House for NinjaTrader 8
VERSION	1.0.0
DATE	November 2026
NINJATRADER	8.1.x Desktop, tested on 8.1.9.0
PACKAGE	`SupremePowerHouse.zip`, compiled NinjaScript Add-On
SHA-256	`392ec5b12168f1b4dd8ec95d0020b558dfc93d1f214bf8966f66f39649f42e14`
AUTHOR	Kestutis Balciunas, https://kestutisbalciunas.com/
PRODUCT PAGE	https://kestutisbalciunas.com/ninjatrader/supreme-power-house/
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#). It finds swing highs and lows, and when a bar closes beyond an unbroken swing it reads a change of character (CHoCH, a break against the current structure) or a break of structure (BOS, a break in its direction). A break becomes a BUY or SELL signal when every enabled filter agrees. Seven checks stand behind the filters and the grade: momentum, the higher-timeframe trend, the lower-timeframe trend, volume, breakout, order flow (CVD) and the trend ribbon. Each signal gets a grade from how many of the seven agree, enabled or not: A from five, B from three, otherwise C.

Each signal draws a trade plan: an entry line, a stop, up to three targets, a risk zone and reward bands. It also draws an adaptive (Kaufman) trend ribbon with matching candle colours, Get Ready plates before a break, the OTE zone and the equilibrium of the current leg, next-trigger levels, a six-timeframe dashboard, and eight alert conditions. It only draws: it never places, modifies or closes orders.

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 with volume for the chart and for the other timeframes the indicator reads: up to 1 minute, up to weekly at the defaults. No Order Flow+ features, no depth, no tick history.

Import the Add-On package

1. Control Center > Tools > Import > NinjaScript Add-On.
2. Select `SupremePowerHouse.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 **Supreme Power House**, click add, click OK.
3. NinjaTrader loads the other timeframes first, so the first picture can take a little longer than on a single-series indicator. The chart then shows the trend ribbon with coloured candles, swing labels, the next BUY and SELL trigger levels, the OTE zone and the equilibrium of the current leg, and the dashboard in the top-right corner. BUY and SELL plates with their trade plans appear where the bars on the chart produced signals. The newest five plans stay drawn.

Uninstall / update

- Remove: Control Center > Tools > Remove NinjaScript Assembly > Supreme Power House.
- 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 **Supreme Power House** to a chart (Ctrl+I). Leave every setting at its default.
2. Load enough history: the swings, the ribbon and the plan statistics are built from the bars the chart loaded, so a short chart can differ for a while from one with more days. **Days to load** is in the chart's Data Series dialog.

Reading the chart

- **BUY and SELL plates**: on the bar that closed beyond an unbroken swing while every enabled filter agreed. The letter is the grade: A when at least five of the seven checks agree, enabled or not, B from three, otherwise C. A stronger grade has a more solid plate. Rest the pointer on a plate to see the signal's kind, entry, stop and the seven checks.
- **Ready plates**: amber, printed when price closes within **within ATR** of an unbroken swing while the enabled filters other than momentum and breakout already agree, once per swing. Rest the pointer on one to see the level it is waiting for. It says price is close to the level, not that it will break.
- **Structure lines**: a dashed line from the broken swing to the breaking bar, labelled CHoCH (against the structure) or BOS (with it).
- **Swing labels**: HH, HL, LH and LL against the previous swing of the same side, printed once the swing is confirmed, **Structure length** bars after it formed. The newest 40 stay.
- **Trade plan**: the entry line, the stop line and the risk zone behind it, a reward band up to each target, and the target lines with their prices. The next target and any reached one are drawn solid and thicker. A reached target is marked with a tick, a reached stop with a cross. A closed plan fades. The newest five plans stay drawn (**Signals kept on chart**).
- **Trend ribbon**: the adaptive average with a two-layer band, green up, red down, fading through grey in between. The band sits below the line in an up direction and above it in a down direction, and the line breaks on a bar where the direction flips. **Candle colours** paints the candles to match.
- **OTE zone**: the box between **OTE from** and **OTE to** of the current leg, measured back from its extreme. The dotted **EQ** line is the leg's midpoint. The upper half of the leg is shaded and labelled Premium, the lower half Discount.
- **Trigger levels**: dotted lines at the unbroken swing high and low, with their prices. A close beyond one is the next structure break.
- **Dashboard**: the bias across the six timeframes in the header. Then one row per timeframe (Strong when the close and its EMA both point that way, Weak when only the close does, n/a below the chart's timeframe), volatility from the ATR percentile, the ribbon's direction and conviction, the latest signal and its state, and the distance to the next BUY and SELL triggers in ATR. Rest the pointer on a cell for its explanation.

The objects past the last candle (triggers, EQ, the zones' right ends, a live plan's labels) sit up to 20 bars ahead. **Pin Labels Inside Chart** compresses that space so they fit. Switch it off and widen chart Properties > Right side margin to see them at their designed distance.

How a plan ends

On every closed bar the live plan is checked first, and the stop before the targets: a bar that touches both counts as stopped. Then the bar is checked for a new break. A new signal ends the previous plan where it stands. With **One signal per direction until the opposite signal** on, the next BUY waits for a SELL, and the other way round.

Settings to try first

SETTING	WHAT IT CHANGES
Structure length	Bars on each side of a swing. Lower gives smaller swings and more signals, higher gives bigger structure.
Signals	Both kinds of break, or only CHoCH, or only BOS.
Higher-timeframe	The timeframe of the higher-timeframe filter, 1 hour by default. It should sit above the chart's timeframe.
02 Filters	Momentum, the higher-timeframe trend and the ribbon are on by default; lower-timeframe trend, volume, breakout, order flow and the session window are off. Every enabled filter must agree.
Stop and **Targets in**	A stop beyond the swing or a fixed ATR multiple; targets as risk multiples, ATR multiples or points.
Dashboard position / text size	Five positions and three sizes.

Alerts

BUY and SELL are on by default; the Get Ready, target and stop alerts are off (**09 Alerts**). An alert fires once, when its bar closes, on a live chart only. It appears in the Alerts Log, for example:

```
BUY signal | Supreme Power House: BUY signal (CHoCH, grade A, entry 20123.25, stop 20098.75) on MNQ 5 @ 20123.25
```

The prices are an example. Sound and e-mail are optional. **Test Alerts on Start** sends one test through every channel you switched on, so you can check them without waiting for a signal. There is no webhook in this edition.

What it does not do

It places no orders. A plate, a grade and a plan describe what the closed bars show; they do not forecast where price goes next. The hit rates on the target labels (**Hit rate on targets**) count levels touched on the bars of this chart: they are no result of a trading test.

04

Calculate mode and bar timing

- Default `Calculate.OnBarClose`: one commit per closed bar. Signals, plans, Get Ready plates, targets and stops are decided on closed bars only, and every alert describes a closed bar.
- `OnEachTick` / `OnPriceChange`: the closed bar is committed once, at the first tick of the next bar, with the same result. In between the forming bar is previewed at most four times a second. A preview moves the ribbon, the candle colour, the OTE zone, the equilibrium, the triggers and the dashboard, and is rolled back completely before the next one, so it leaves no trace in the signals.
- With `OnBarClose`, the chart shows the state as of the last closed bar. The dashboard's matrix rows read the timeframe bar that had closed when the newest committed bar opened, so a higher-timeframe roll shows one chart bar later than under the per-tick modes.
- When the chart loads, a still-forming last bar is previewed, not committed.
- With no tick arriving, a bar has not closed for NinjaTrader. On a closed market the session's last bar stays a preview until the first tick after the reopen and is committed then. See the chapter on sessions, time and volume.
- A bar's OPEN time is its NinjaTrader close time minus the bar period; on day-based bars it is the end of the previous bar.
- Other timeframes never read ahead. See the chapter on sessions, time and volume.

05

Outputs for strategies and the Market Analyzer

PLOT / SERIES	VALUES
`Ribbon` (plot 0, drawn)	the adaptive average; no value on a bar where the ribbon flips, or with Ribbon off
`RibbonInnerEdge` (plot 1, invisible)	the band's inner edge; no value where the ribbon has none
`RibbonOuterEdge` (plot 2, drawn)	the band's outer edge; no value where the ribbon has none
`BuySignal`, `SellSignal` (plots 3, 4, hidden)	1 on the closed bar of the signal, otherwise 0
`GetReadyBuy`, `GetReadySell` (plots 5, 6, hidden)	1 on the closed bar of the Get Ready plate, otherwise 0
`Tp1Hit`, `Tp2Hit`, `Tp3Hit`, `StopHit` (plots 7 to 10, hidden)	1 on the closed bar the live plan reached that level, otherwise 0

A forming bar reads 0 on the eight events: every one of them is decided when the bar closes. The eight hidden series follow the eight alert conditions and fire whatever the alert switches say.

06

Parameters

The 72 inputs in groups 01 to 07 keep the titles, order, defaults and ranges of the other editions (the group names carry a number prefix because NinjaTrader sorts groups alphabetically; a timeframe is chosen from a list). Groups 08 Display, 09 Alerts and 10 NinjaTrader hold the 18 platform settings: 90 parameters in total.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
01 Signals	Signals	NinjaTrader.NinjaScript.SphSignalMode	NinjaTrader.NinjaScript.SphSignalMode.CHoCHAndBOS		`SigMode`	CHoCHAndBOS: both kinds of break can become signals. CHoCHOnly: only a break against the current structure (a possible reversal). BOSOnly: only a break in its direction (continuation). The structure lines are drawn for both either way.
01 Signals	Structure length	int	5	2 to 50	`PvLen`	Bars on each side of a swing high or low; a swing is confirmed this many bars after it formed. Lower = faster signals, higher = bigger structure.
01 Signals	One signal per direction until the opposite signal	bool	true		`OneDir`	After a BUY, the next BUY waits for a SELL, and the other way round. Get Ready plates follow the same rule.
01 Signals	Get Ready pre-signals	bool	true		`ShowReady`	Prints a Get Ready plate when price closes within the distance below of an unbroken swing while the enabled filters other than momentum and breakout already agree. Once per swing.
01 Signals	within ATR	double	0.5	0.05,double.MaxValue	`ReadyK`	Price is within this many ATR (14) of the level that would trigger a signal.
01 Signals	Show grade on signals	bool	true		`ShowGrade`	Adds the grade (A, B or C) to the BUY and SELL plates. A: at least five of the seven checks agree, enabled or not. B: at least three. The grade is computed either way and stays in the tooltip and the alerts.
01 Signals	Structure lines (CHoCH / BOS)	bool	true		`ShowStr`	Draws a dashed line from the broken swing to the breaking bar, labelled CHoCH or BOS. Drawing only.
01 Signals	Swing labels (HH / HL / LH / LL)	bool	true		`ShowSw`	Labels each confirmed swing against the previous one of its side. Drawing only; the newest 40 stay.
01 Signals	Next BUY / SELL trigger levels	bool	true		`ShowTrig`	Dotted lines at the unbroken swing high and low: a close beyond them is the next structure break. Drawing only.
02 Filters (every enabled filter must agree)	Momentum	bool	true		`FMom`	A BUY needs the close to have risen at least the Momentum ATR distance over the Momentum bars; a SELL the mirror.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
02 Filters (every enabled filter must agree)	Momentum bars	int	5	1,int.MaxValue	`MomLen`	The bars the momentum is measured over.
02 Filters (every enabled filter must agree)	Momentum ATR	double	0.5	0.0,double.MaxValue	`MomK`	The minimum move, in ATR (14), the momentum filter requires.
02 Filters (every enabled filter must agree)	Higher-timeframe trend	bool	true		`FHtf`	A BUY needs the last closed Higher-timeframe bar to have closed above its Higher-timeframe EMA; a SELL below.
02 Filters (every enabled filter must agree)	Higher-timeframe	NinjaTrader.NinjaScript.SphTimeframe	NinjaTrader.NinjaScript.SphTimeframe.Hour1		`HtfTf`	The timeframe of the higher-timeframe filter. It loads 5,000 bars at most, so a chart reaching further back reads no trend on its oldest bars.
02 Filters (every enabled filter must agree)	Higher-timeframe EMA	int	50	2,int.MaxValue	`HtfLen`	The EMA length on the higher timeframe.
02 Filters (every enabled filter must agree)	Lower-timeframe trend	bool	false		`FLtf`	A BUY needs the latest Lower-timeframe close above its Lower-timeframe EMA; a SELL below. Off by default.
02 Filters (every enabled filter must agree)	Lower-timeframe	NinjaTrader.NinjaScript.SphTimeframe	NinjaTrader.NinjaScript.SphTimeframe.Minute1		`LtfTf`	The timeframe of the lower-timeframe filter. Its bars load whether the filter is on or not, because the check counts towards the grade.
02 Filters (every enabled filter must agree)	Lower-timeframe EMA	int	20	2,int.MaxValue	`LtfLen`	The EMA length on the lower timeframe.
02 Filters (every enabled filter must agree)	Volume	bool	false		`FVol`	The signal bar's volume must be at least Volume × average times its 20-bar average. Off by default.
02 Filters (every enabled filter must agree)	Volume × average	double	1.2	0.1,double.MaxValue	`VolK`	The volume multiple the volume filter requires.
02 Filters (every enabled filter must agree)	Breakout	bool	false		`FBrk`	The signal candle also closes beyond the highest high (lowest low) of the lookback. Off by default.
02 Filters (every enabled filter must agree)	Breakout bars	int	20	2,int.MaxValue	`BrkLen`	The lookback of the breakout filter.
02 Filters (every enabled filter must agree)	Order flow (CVD)	bool	false		`FFlow`	Cumulative volume delta over the lookback points the same way. The delta is estimated from lower-timeframe candles, up candles counted as buying and down candles as selling; a bar without them is split by where its close sits in its range. Off by default.
02 Filters (every enabled filter must agree)	Order flow bars	int	20	2,int.MaxValue	`FlowLen`	The lookback of the order-flow filter.
02 Filters (every enabled filter must agree)	Trend ribbon	bool	true		`FRib`	Blocks a signal only when the ribbon's trend score clearly points the other way: a BUY at or below -0.2, a SELL at or above +0.2. A neutral (grey) ribbon allows signals in both directions, so reversals (CHoCH) are not filtered out.
02 Filters (every enabled filter must agree)	Session window	bool	false		`FSes`	Allows signals and Get Ready plates only inside Session. Off by default.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
02 Filters (every enabled filter must agree)	Session	string	"0700-1600"		`SesStr`	The window as HHMM-HHMM in Session timezone: the start is included, the end is not, every day.
02 Filters (every enabled filter must agree)	Session timezone	string	"America/New_York"		`SesTz`	An IANA zone such as America/New_York or Europe/London; daylight saving is followed. An unknown zone falls back to UTC and prints a line in the NinjaScript Output window.
03 Trend ribbon	Ribbon	bool	true		`ShowRib`	Draws the adaptive (Kaufman) moving average with its two-layer band, coloured by the trend score: green up, red down, fading through grey in between. Drawing only; the trend-ribbon filter works either way.
03 Trend ribbon	Candle colours	bool	true		`ShowBar`	Paints each candle in the ribbon's colour. Your own candle colours return when this is off or the indicator is removed.
03 Trend ribbon	Adaptive length	int	10	2,int.MaxValue	`ErLen`	Bars of the efficiency ratio: the straighter price has moved over them, the faster the average follows.
03 Trend ribbon	Adaptive slow	int	30	3,int.MaxValue	`SlowLen`	The slowest the average follows in choppy price, as an EMA length.
03 Trend ribbon	Flip band (ATR)	double	0.6	0.0,double.MaxValue	`RibBand`	The ribbon only changes direction when price closes this many ATR beyond the baseline and the trend score agrees. Higher = fewer flips in consolidation.
03 Trend ribbon	Colour smoothing (bars)	int	10	1 to 50	`RibSm`	The colour follows a smoothed trend score: it fades through neutral grey when the market goes sideways instead of jumping between red and green.
04 Trade plan	Stop	NinjaTrader.NinjaScript.SphStopType	NinjaTrader.NinjaScript.SphStopType.Structure		`SlType`	Structure: beyond the newest opposite swing plus 0.1 ATR, kept between the min and max ATR distance. ATR: a fixed ATR multiple from the entry.
04 Trade plan	Stop ATR ×	double	1.5	0.1,double.MaxValue	`SlAtr`	The stop distance in ATR for the ATR stop, and for the Structure stop before the first swing exists.
04 Trade plan	Stop distance (ATR): min	double	0.5	0.1,double.MaxValue	`MinRisk`	The closest a Structure stop may be to the entry, in ATR.
04 Trade plan	Stop distance (ATR): max	double	3.0	0.2,double.MaxValue	`MaxRisk`	The farthest a Structure stop may be from the entry, in ATR.
04 Trade plan	Targets in	NinjaTrader.NinjaScript.SphTargetMode	NinjaTrader.NinjaScript.SphTargetMode.RMultiple		`TpMode`	RMultiple: the TP values are multiples of the risk (entry to stop). ATR: multiples of the ATR. Points: price distances.
04 Trade plan	TP1	bool	true		`Tp1On`	The first target. Off removes it from the plan and its drawing.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
04 Trade plan	TP1 value	double	1.0	0.01,double.MaxValue	`Tp1`	TP1's distance from the entry, in the unit of Targets in.
04 Trade plan	TP2	bool	true		`Tp2On`	The second target. Off removes it from the plan and its drawing.
04 Trade plan	TP2 value	double	2.0	0.01,double.MaxValue	`Tp2`	TP2's distance from the entry, in the unit of Targets in.
04 Trade plan	TP3	bool	true		`Tp3On`	The third target. Off removes it from the plan and its drawing.
04 Trade plan	TP3 value	double	3.0	0.01,double.MaxValue	`Tp3`	TP3's distance from the entry, in the unit of Targets in.
04 Trade plan	Show trade plan	bool	true		`ShowTP`	Draws each signal's entry and stop lines, the risk zone, the reward bands and the target lines with their labels. Off keeps the plates; the plan is still tracked for the alerts and the dashboard.
04 Trade plan	Hit rate on targets	bool	false		`ShowPct`	Adds to each target label the share of closed plans that reached it, once at least 20 plans have closed. Off by default.
04 Trade plan	Plan length (bars)	int	20	5 to 200	`ProjBars`	How far right each plan is drawn from its signal bar; a live plan reaches at least three bars past the last bar.
04 Trade plan	Signals kept on chart	int	5	1 to 50	`KeepSig`	How many of the newest plans stay drawn. Structure lines keep four times as many, Get Ready plates twice as many.
05 ICT	OTE zone of the current leg	bool	true		`ShowOTE`	Shades the optimal trade entry box of the current leg, between OTE from and OTE to of the leg measured back from its extreme.
05 ICT	OTE from	double	0.62	0.0 to 1.0	`OteLo`	The retracement fraction where the OTE box starts.
05 ICT	OTE to	double	0.79	0.0 to 1.0	`OteHi`	The retracement fraction where the OTE box ends.
05 ICT	Equilibrium (50%)	bool	true		`ShowEQ`	Draws the leg's midpoint as a dotted line labelled EQ.
05 ICT	Premium / discount shading	bool	true		`ShowPD`	Shades the half of the leg above the equilibrium in the sell colour and the half below in the buy colour. Drawn while the OTE zone or the equilibrium is on.
06 Dashboard	Dashboard	bool	true		`ShowDash`	Shows the table: the bias, the six-timeframe matrix, volatility, the ribbon, the latest signal and the next triggers. Hover a cell for its explanation.
06 Dashboard	Dashboard position	NinjaTrader.NinjaScript.SphDashPosition	NinjaTrader.NinjaScript.SphDashPosition.TopRight		`DashPos`	Where the table sits.
06 Dashboard	Dashboard text size	NinjaTrader.NinjaScript.SphDashSize	NinjaTrader.NinjaScript.SphDashSize.Small		`DashSz`	Tiny, Small or Normal.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
06 Dashboard	Matrix 1	NinjaTrader.NinjaScript.SphTimeframe	NinjaTrader.NinjaScript.SphTimeframe.Minute5		`Mtf1`	The first matrix timeframe. A timeframe below the chart's reads n/a.
06 Dashboard	Matrix 2	NinjaTrader.NinjaScript.SphTimeframe	NinjaTrader.NinjaScript.SphTimeframe.Minute15		`Mtf2`	The second matrix timeframe.
06 Dashboard	Matrix 3	NinjaTrader.NinjaScript.SphTimeframe	NinjaTrader.NinjaScript.SphTimeframe.Hour1		`Mtf3`	The third matrix timeframe.
06 Dashboard	Matrix 4	NinjaTrader.NinjaScript.SphTimeframe	NinjaTrader.NinjaScript.SphTimeframe.Hour4		`Mtf4`	The fourth matrix timeframe.
06 Dashboard	Matrix 5	NinjaTrader.NinjaScript.SphTimeframe	NinjaTrader.NinjaScript.SphTimeframe.Daily		`Mtf5`	The fifth matrix timeframe.
06 Dashboard	Matrix 6	NinjaTrader.NinjaScript.SphTimeframe	NinjaTrader.NinjaScript.SphTimeframe.Weekly		`Mtf6`	The sixth matrix timeframe.
06 Dashboard	Matrix EMA	int	50	2,int.MaxValue	`MtfLen`	The EMA length on every matrix timeframe: the close above it reads an up triangle, below a down triangle; Strong when the EMA also points that way.
07 Colours	Buy	colour	#089981		`BullBrush`	The bullish colour: BUY plates, the rising ribbon, the buy trigger, the discount shading and bullish dashboard cells.
07 Colours	Sell	colour	#F23645		`BearBrush`	The bearish colour: SELL plates, the falling ribbon, the sell trigger, the upper half's shading and bearish dashboard cells.
07 Colours	Get Ready	colour	#F5A623		`ReadyBrush`	Get Ready plates.
07 Colours	CHoCH	colour	#5B9CF6		`ChochBrush`	CHoCH lines and labels.
07 Colours	BOS	colour	#9598A1		`BosBrush`	BOS lines and labels.
07 Colours	Targets	colour	#26A69A		`TpBrush`	Target lines, labels and reward bands.
07 Colours	Stop colour	colour	#EF5350		`SlBrush`	The stop line, its label and the risk zone.
07 Colours	Entry	colour	#B2B5BE		`EntBrush`	The entry line.
08 Display	Keep Chart Object Limits	bool	true		`KeepChartObjectLimits`	On: the newest 500 boxes, 500 lines and 500 labels stay on the chart and older drawings are retired, as the design intends. Off: every drawing stays.
08 Display	Pin Labels Inside Chart	bool	true		`PinLabelsInside`	The trigger levels, the equilibrium, the zones' right ends and a live plan's SL / TP labels sit up to 20 bars right of the last candle. On: while the last candle is on screen, the space after it is compressed so they fit inside the chart, in their designed order. Off: they stay at their designed position; widen the chart's right side margin to see them.
09 Alerts	Alert: BUY signal	bool	true		`AlertBuy`	Alerts on each BUY. Every alert appears in the Alerts Log window and fires once, when its bar closes.
09 Alerts	Alert: SELL signal	bool	true		`AlertSell`	Alerts on each SELL.
09 Alerts	Alert: Get Ready BUY	bool	false		`AlertReadyBuy`	Alerts on each Get Ready BUY plate.
09 Alerts	Alert: Get Ready SELL	bool	false		`AlertReadySell`	Alerts on each Get Ready SELL plate.
09 Alerts	Alert: TP1 hit	bool	false		`AlertTp1`	Alerts when the live plan reaches TP1.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
09 Alerts	Alert: TP2 hit	bool	false		`AlertTp2`	Alerts when the live plan reaches TP2.
09 Alerts	Alert: TP3 hit	bool	false		`AlertTp3`	Alerts when the live plan reaches TP3.
09 Alerts	Alert: Stop hit	bool	false		`AlertStop`	Alerts when the live plan reaches its stop.
09 Alerts	Alert Log Line	bool	true		`AlertLogLine`	Writes each alert to the NinjaScript Output window.
09 Alerts	Alert Sound	bool	false		`AlertSound`	Plays a sound with each alert.
09 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.
09 Alerts	Alert E-mail	bool	false		`AlertEmail`	Sends each alert by e-mail through the Mail share service configured in NinjaTrader (Tools > Options > Share).
09 Alerts	E-mail To	string	""		`EmailTo`	Recipient address for alert e-mails. Blank sends none.
09 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.
10 NinjaTrader	Debug Mode	bool	false		`DebugMode`	Writes diagnostics to sph_nt_debug.log in the NinjaTrader 8 folder.
10 NinjaTrader	Dump Bars + Outputs CSV	bool	false		`DebugDumpCsv`	Writes every committed bar with the engine outputs to sph_dump_<instrument>_<period>.csv, and the bars of the other timeframes beside it (parity checks).

90 parameters.

07

Sessions, time and volume

Sessions

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 ribbon, the ATR and the signals; with an RTH template they do not. Both are valid; the values differ, as they do for any indicator.

The other timeframes are requested for the chart's instrument, with the template left to NinjaTrader's default for such a request. A chart on an RTH template may read other timeframes on the instrument's default template rather than the chart's.

The Session window

The optional **Session window** filter is separate from the chart's template. It allows signals and Get Ready plates only when the bar's open falls inside **Session**, written `HHMM-HHMM` in **Session timezone**: the start is included, the end is not, every day, and a window can wrap midnight. Use a zone name such as `America/New_York` or `Europe/London`; daylight saving is followed. A zone the indicator does not know falls back to UTC, and a line in the NinjaScript Output window says so. An invalid **Session** text matches nothing, so with the filter on no signal prints.

The final bar of a session

With no tick arriving, a bar has not closed for NinjaTrader. On a closed market the session's last bar stays a preview until the first tick after the reopen, and it is committed then. So a signal on that bar appears at the reopen, and its alert is raised only if the reopen comes within two bar periods plus five seconds of that bar's close; otherwise the signal prints with no alert. The same applies to any later close on a quiet instrument.

History

The swings, the ribbon and the plans 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. The newest 500 to 505 boxes, lines and labels stay on the chart, and the oldest are retired first (**Keep Chart Object Limits**).

Other timeframes

Each secondary series loads its own bars, and you get what your feed holds:

WHAT READS IT	BARS REQUESTED
Higher-timeframe filter	5,000 (a weekly timeframe 520, a monthly one 240)
A matrix row	20 x **Matrix EMA**, from 500 to 5,000: 1,000 at the defaults (a weekly row 520, a monthly one 240)
Lower-timeframe filter and the order-flow intrabars, on a chart below one hour	the chart's own data range
Lower-timeframe filter and the order-flow intrabars, on a chart of one hour or more	the newest 100,000 bars

A timeframe equal to the chart's reads the chart's own bars. Two requests for one timeframe share a series. So a chart reaching back further than 5,000 bars of the higher timeframe reads no higher-timeframe trend on its oldest bars, and with that filter on it prints no signal there.

No read goes ahead of the bar that made it, on history and live alike:

- **The higher-timeframe filter and the matrix** read the newest bar of their timeframe that had ended when the chart bar opened. A chart bar inside the 10:00 hour reads the 09:00 hour's bar. A session's shortened last bar counts as ended when it really ended.
- **The lower-timeframe filter** reads, on a closed bar, the newest bar that had ended by the chart bar's close. Only a forming chart bar reads a forming lower-timeframe bar, and a forming bar decides nothing.
- **A daily, weekly or monthly bar** ends at the session end of its last trading date, taken from NinjaTrader's session information, not at the midnight of its date stamp.

A dashboard row is only as current as NinjaTrader's series of that timeframe. A daily or weekly series can lag or have holes on some connections while the minute bars are complete; this was measured on a verification PC, where a daily series had no bars for a whole trading week. The row then reads the newest bar the series holds, with no message. **Debug Mode** writes each series' asked and loaded bar count and its first and last bar end to `sph_nt_debug.log`.

Time stamps

NinjaTrader stamps an intraday bar with its CLOSE time in the platform time zone. The indicator converts it to UTC through the zone chosen under Tools > Options, and a chart bar's open time is that close minus the bar period. A daily bar carries its trading date at midnight, a weekly bar the Friday of its week and a monthly bar the last day of its month, so such a bar's end comes from the session information and its open is the end of the bar before it. Everything drawn is positioned by bar index. Slots past the last bar, where the plans, the triggers and the zones end, are placed at the average bar spacing of the view, or closer with **Pin Labels Inside Chart** on.

When the clocks go back, an hour of stamps repeats. The readings of that hour are older by up to an hour, and never newer. Futures sessions are closed at that time.

Volume

Volume on a futures chart is real contract volume. It is read in two places:

- **The volume check**: the bar's volume against its 20-bar average, against **Volume × average**.
- **The order-flow check (CVD)**: an estimate, not tick data. Each chart bar's delta comes from the candles of a lower timeframe that end inside it: an up or flat candle's volume counts as buying, a down candle's as selling. The lower timeframe follows the chart: 1 minute up to a 15-minute chart, 5 minutes up to 1 hour, 15 minutes up to 4 hours, 1 hour up to daily, 4 hours above. On a 1-minute chart the bar is its own single candle. A bar with no candles (older than the lower series reaches, a chart below 1 minute, or a non-time chart) is split by where its close sits in its range.

Both checks count towards the grade even with their filters off. A different feed gives different volume, so grades can differ between feeds on the same symbol.

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, the ribbon and the ATR behind the stops are built from prices, so a non-back-adjusted roll can produce a false break and an oversized ATR for a while, and the same holds for each other timeframe on its own bars. 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, with secondary data series of the chart's own instrument: six at the defaults on a 1 or 5 minute chart. A timeframe equal to the chart's reads the chart's own bars, a matrix row below the chart's timeframe reads nothing and prints n/a.
Instruments	Any instrument with bars and volume: futures, micro futures, forex, stocks. Prices print at the decimals of the instrument's tick, quarter ticks included.
Bar types	Time-based (second, minute, day, week, month, year). Tick, volume, range, Renko and other non-time bars run: their order-flow delta always uses the candle split, the chart's timeframe is taken from the smallest bar spacing of the first 300 bars, and the alert age limit is five minutes, or two of those spacings plus five seconds, whichever is longer.
Timeframe inputs	A list of 26: 1 to 10, 15, 20, 30 and 45 minutes, 1, 2, 3, 4, 6, 8 and 12 hours, 1, 2 and 3 days, 1 week, 1 month.
Calculate modes	`OnBarClose` (default), `OnEachTick`, `OnPriceChange` (forming bar previewed at most four times a second).
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.
Data requirements	Bars with volume, of the chart and of the other timeframes, daily and weekly included. A timeframe with no closed bar, or too few to seed its EMA, reads `—` on the dashboard and reads no trend in the filter. With the higher-timeframe filter on, no signal can print until its timeframe has a trend. On a feed without volume the volume check does not agree and the order-flow delta stays flat.
Load and speed	Every tick updates the secondary series as well. Chart intervals from one hour up load up to 100,000 lower-timeframe bars.
Multi-instance	Yes. Every instance paints its own surface. Two dashboards at one position overlap: use **Dashboard position**.
Market Analyzer	Not tested. The code draws nothing without a chart and the series in the chapter on outputs are plots, but each instrument loads the secondary series too. Not advertised as a scanner.
Strategies	Not tested. The series are plots and every input except the colours is a `[NinjaScriptProperty]`. NinjaTrader expects a calling strategy to add the same secondary series; a series the strategy has not added stays empty.
Rendering	SharpDX in `OnRender` with DPI scaling and full resource cleanup on render-target change and termination. No draw objects. The three ribbon lines are NinjaTrader's own plots.
Candle colours	`BarBrushes` and `CandleOutlineBrushes`, on candlestick and other bar styles the chart supports. Your own colours return when **Candle colours** is off or the indicator is removed.
Dark and light charts	Buy, sell, target, stop, entry and structure colours are inputs. The dashboard's own colours, the swing labels and the tooltip are fixed, chosen for a dark chart.
Templates and workspaces	All settings serialize (colours through the standard brush serializer).
Minimized chart	Keeps calculating and alerting while its chart window is minimized. 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.
Alerts	Alerts Log, NinjaScript Output, sound, and e-mail through NinjaTrader's Mail share service. No webhook, and no network call of its own.
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.
The chart takes a while to appear	NinjaTrader loads the other timeframes first: six series at the defaults on a 1 or 5 minute chart, up to 100,000 lower-timeframe bars on a chart of one hour or more. The first load from the data provider is the slowest.
No BUY or SELL plates	Every enabled filter must agree, and a timeframe with no data reads no trend. Check the dashboard's matrix for `—`, then switch **Higher-timeframe trend** off to see whether that filter blocks the signals. **One signal per direction until the opposite signal** also holds back a repeat.
No signal on the oldest bars of a long chart	The higher-timeframe series loads 5,000 bars and the filter reads no trend before they start. See the chapter on sessions, time and volume.
A matrix row reads n/a	Its timeframe is below the chart's. Pick a higher one in **06 Dashboard**.
A matrix row reads `—` or looks old	That timeframe has too few closed bars for its EMA, or the feed's series lags. See the chapter on sessions, time and volume. Reload the chart's data (right-click > Reload All Historical Data).
The plates update only when a bar closes	That is the design: signals, plans and alerts belong to closed bars. **Calculate** `OnEachTick` or `OnPriceChange` also moves the ribbon, the candle colour, the zones and the dashboard on the forming bar.
A signal appears only at the reopen	NinjaTrader closes a bar when the next bar's first tick arrives. See the chapter on sessions, time and volume.
No alert on a signal	BUY and SELL are on by default, the others are off (**09 Alerts**). An alert fires on a live chart only, once when its bar closes, and not for a bar that closed more than two bar periods plus five seconds ago (at least five minutes on tick, volume and range bars).
Candles keep their colour	**Candle colours** is off, or these are the first bars, before the trend score exists.
The ribbon line has a gap	The line breaks on a bar where the ribbon changes direction. That is the design.
Labels past the last candle are squeezed or off the chart	**Pin Labels Inside Chart** compresses the space after the last candle so they fit. With it off they stay at their designed distance: widen chart Properties > Right side margin.
The dashboard covers something	Choose another **Dashboard position**, a smaller **Dashboard text size**, or switch **Dashboard** off.
The session window blocks everything	**Session** must read `HHMM-HHMM`. An invalid text matches nothing. A line in the NinjaScript Output window names a zone the indicator does not know; it then uses UTC.
An e-mail alert does not arrive	**E-mail To** must be filled, and the Mail share service set up in NinjaTrader (Tools > Options > Share). A failure prints one line in the NinjaScript Output window.
Values differ from another platform on the same symbol	Bars and volume differ by feed and session template, and the volume and order-flow checks feed the grade; compare on identical bars (see the chapter on how this edition was verified).
Text is tiny or huge	Label and table sizes follow the Windows display scaling and **Dashboard text size**.
An error line names Supreme Power House in the Log tab	Set **10 NinjaTrader > Debug Mode** on, reproduce it, and send `Documents\NinjaTrader 8\sph_nt_debug.log` with the chart interval, instrument and NinjaTrader version.
Two dashboards on one chart	Two instances are configured (Ctrl+I, Configured list). Remove one.

A daily row, a VWAP or a previous-day level looks wrong or stops updating

NinjaTrader can keep using an older copy of its daily or minute history on your PC, and reloading the chart does not refresh it. Open **Control Center > Tools > Historical Data**, expand **Download**, pick the instrument, select **Day**, set the dates to cover the missing days and press **Download**. Do the same with **Minute** if intraday bars are missing, then reload the chart.

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 runs 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 and compared with reference values for the same bars, value by value and drawn object by drawn object:

SCOPE	COMPARED	MISMATCHES
Calculated values, colours and the eight alert conditions, per bar, in 152 runs on EURUSD 5, 15 and 60 minutes and XAUUSD 5 minutes, including input sweeps	73,115,364 cells	0
Drawn objects: every label, line, box and table cell, in the same runs	79,409 fields	0

Two items are declared in that comparison. The lower-timeframe average was seeded from a reference value at its first bar, because its reference series starts before the first bar any capture holds; every value after it is computed and compared. On the 15 and 60 minute charts the matrix rows below the chart's timeframe were not compared: they print n/a and reach no output.

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 ribbon's calculation and colours, the filters' inputs, the requests of the other timeframes, the swing and structure state, the filter and grade results, the live plan and its counters, the eight alert conditions. It also writes the bars of each other timeframe and the intrabars of each chart bar. 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 1 Minute, historical load, NinjaTrader 8.1.9.0, default inputs	4,923	103	507,069	0
MNQ 1 Minute, across the switch from history to live bars, Calculate = OnEachTick	4,191	103	431,673	0

Bar times matched on every row. The bars run from 27 October 22:00 UTC to 2 November 2026 12:02 UTC, overnight hours included. On this run the engine produced 2 BUY and 2 SELL signals, 2 Get Ready plates, 3 target events and 4 stop events, on the same bars on both sides, so the signal columns were exercised on those few bars only. Both sides read the same bars of the other timeframes, so this stage shows that the host and the engine agree, not that a feed's series are complete: see the chapter on sessions, time and volume.

Complete daily data, 2 Nov 2026: after the daily and minute history were downloaded, a further live run had no holes (5,371 bars, 27 October to 2 November) and the daily series delivered new bars for 26 to 30 October: 103 series, 553,213 cells, 0 mismatches, 0 bar-identity mismatches. The D and W rows read the bars of 30 October.

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 **10 NinjaTrader > Debug Mode** on, reproduce it once and attach `sph_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/supreme-power-house/. 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.

Supreme Power House is a technical analysis tool. It draws structure breaks, a trend ribbon and level plans computed from historical bars; it does not place, modify or close orders and it gives no trade advice. The grades, the checks and the hit rates describe the bars on the chart; they are descriptions, not probabilities and not forecasts. 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.