Kestutis Balciunas

NINJATRADER 8 INDICATOR · NINJASCRIPT, C#

User Guide

A rolling volume profile with POC, value area, peaks, troughs and ranked rows that become key levels, closed-bar break, rejection and retest signals, a metrics table and alerts.

PRODUCT	Supreme Dynamic Profile for NinjaTrader 8
VERSION	1.0.0
DATE	November 2026
NINJATRADER	8.1.x Desktop, tested on 8.1.9.0
PACKAGE	`SupremeDynamicProfile.zip`, compiled NinjaScript Add-On
SHA-256	`70cf1e932c0b676c7be4ce9b6838341011 32cde664ca959c80b1e496f2743325`
AUTHOR	Kestutis Balciunas, https://kestutisbalciunas.com/
PRODUCT PAGE	https://kestutisbalciunas.com/ninjatrader/supreme-dynamic-profile/
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#): a rolling volume profile of the most recent closed bars whose Point of Control (POC), value area, volume peaks, volume troughs and ranked rows become key levels on the chart. It spreads each bar's volume over the price rows the bar covered, draws the latest profile as a histogram, keeps the levels running to the right, and marks a closed bar that breaks a level, is rejected by it, or retests it after a break. A metrics table reports where price sits against the value area and the nearest levels. It has nine alert conditions and ten series a strategy can read.

The volume is estimated volume at price: it comes from bars, not from exchange bid and ask data. The percentages on the labels describe the profile when the level was created; they are not win probabilities.

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 with volume for the chart. A volume profile is built from volume. No Order Flow+ features, no depth.
- **Use Lower Timeframe** is optional. It needs minute bars of the instrument, and with **Seconds** on, second bars, which NinjaTrader builds from tick data that not every connection delivers for history.

Import the Add-On package

1. Control Center > Tools > Import > NinjaScript Add-On.
2. Select `SupremeDynamicProfile.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 Dynamic Profile**, click add, click OK.
3. The chart shows the profile as a histogram of coloured rows at the right, kept inside the chart, the value-area high and low as lines, volume peaks as shaded zones, one high-volume row as a line, a label on each visible level, the metrics table in the top-right corner, and triangle markers on the bars where a level was broken or rejected.

The POC line is off by default: switch on **01 Components > Point of Control** to add it.

Uninstall / update

- Remove: Control Center > Tools > Remove NinjaScript Assembly > Supreme Dynamic Profile.
- 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 Dynamic Profile** to a chart (Ctrl+I). Leave every setting at its default.
2. Load enough history. The first profile needs at least ten closed bars and is built only inside the newest **Calculation History** bars (2,000 by default), and the levels on screen were built from the bars before them. The indicator does not extend the chart's data: the Data Series setting **Days to load** decides. Aim for at least 2,500 bars where your feed allows it.
3. The profile, the levels' life and the signals are decided on closed bars. With the default `OnBarClose` the chart moves when a bar closes; with `OnEachTick` the level colours, the labels and the table also follow the forming bar.

Reading the chart

- **Profile**: each row is one slice of price, and its length shows how much volume traded there compared with the busiest row. By default the histogram covers the last 240 closed bars in 64 rows and is rebuilt every 5 bars. **Directional** colours a row by the estimated up and down volume in it: greener for more up volume, redder for more down volume. Row length is volume, not buying or selling pressure.
- **Placement**: at the default **Right** the histogram is drawn past the last bar, and **Pin Profile Inside Chart** moves it left just enough to stay inside the chart while the last bar is on screen, so it can overlap the newest candles. With the last bar scrolled away nothing is pinned.
- **Value area (VAH and VAL)**: the band of rows around the busiest row that holds 70 % of the volume. Its two edges are key levels.
- **POC**: the busiest row. Off by default; **Point of Control** adds it, drawn in the chart's text colour.
- **Peaks**: rows busier than their neighbours, drawn as zones by default. At most two per profile.
- **Troughs**: rows quieter than their neighbours, off by default.
- **Ranked rows**: the busiest row that is not already a level (**Highest Rows**, one by default) and, if you ask for them, the quietest (**Lowest Rows**).
- **Labels**: the column of labels on the visible levels: a symbol, the name and a metric, for example `● PEAK 7.3% VOL`. ◆ is the POC, ● a peak or a high-volume row, ○ a trough or a low-volume row, — a value-area edge. The percentage is fixed when the level is created. Resting the pointer on a label shows its statistics.
- **Colours**: a level turns green when a bar closes above it and red when a bar closes below it, and it is neutral until either has happened. The POC keeps the chart's text colour. Levels near price are drawn stronger (**Emphasize Nearby Levels**).
- **Visible levels**: **Visible Active Nodes** limits how many peaks, troughs and ranked rows are drawn: the nearest of each family, then the nearest others. Levels beyond the limit stay tracked and can still alert.
- **Metrics table**: the header `Supreme | DYNAMIC PROFILE`, then where price sits against the value area, its distance from the POC in ATRs, the nearest visible level above and below, and relative volume. **Table Variant = Detailed** adds six rows: the profile's window and age, its volume, the sampling (chart candles or lower-timeframe coverage), the up and down estimate, the active and stored level counts and the last signal.

Signal rules in one paragraph

Each active level is checked once per closed bar, from the bar after it was created. A bar **touches** a level when its range reaches it, and each new contact starts one episode. A **break** is a close on the other side of the level from where price last closed. A **rejection** is a close back on the side price came from, after a touch. A **retest** is a touch and a close on the break's side within **Retest Window** bars of a break. It needs **Stop Extending On Break** off, because at the default a level stops at its break. A bar closing above the level is bullish, one closing below it is bearish. On one level, a break takes priority over a retest and a retest over a rejection. An event is accepted only on a level type **Signal Levels** allows (at the default peaks, troughs and ranked rows, not the POC or the value area), with **Minimum RVOL** met, and at least **Signal Cooldown** bars after that level's last accepted signal. A new level cannot signal on the bar that created it.

Markers

MARKER	EVENT
▲ under the bar	bullish break
▼ over the bar	bearish break
△ under the bar	bullish rejection
▽ over the bar	bearish rejection
◆ under or over the bar	retest, bullish or bearish

At most one marker per direction per bar: a break takes priority over a retest, and a retest over a rejection. **Marker Variant = Markers + Text** adds `BREAK`, `RETEST` or `REJECT`. Resting the pointer on a marker shows every level it fired on.

Settings to try first

SETTING	WHAT IT CHANGES
Lookback Mode	Fixed uses **Lookback Limit** bars. Dynamic starts the profile at the latest confirmed pivot, within that limit, and also rebuilds on a new pivot.
Refresh Every N Bars	1 rebuilds the profile on every closed bar.
Point of Control and **POC Variant**	Adds the POC. Developing draws its path as a step line and creates no fixed level and no signal.
Value Area Variant	Historical keeps stopped lines. Developing draws the two edges as step lines and creates no fixed level and no signal.
Peak Variant, **Trough Variant**, **Ranked Row Variant**	Lines or zones. **Zone Radius (Rows)** sets a zone's width.
Stop Extending On Break	Off keeps a level alive after its break, so retests can print.
Signal Levels	All Levels adds the enabled POC and value-area levels to the signals.
Minimum RVOL	Accepts only events on a bar with this multiple of the preceding average volume.
Contrast, **Accent Variant**	Stronger or softer drawings. Custom reads **11 Custom Transparency**. **Visual Accents** off gives plain drawings.
Use Lower Timeframe	Second and minute charts. Builds the profile from lower-timeframe candles where they exist, and the Detailed table reports the coverage.

Alerts

Nine conditions, each behind its own switch in **13 Alerts**. 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	DEFAULT	FIRES WHEN	MESSAGE
Alert: New Profile Key Level	off	a profile rebuild creates at least one new level	`New Profile Key Level on MNQ 5 @ 20130.50`
Alert: Bullish Key Level Break	on	a close above a level price was under	`Bullish Key Level Break (Peak 20123.25) on MNQ 5 @ 20130.50`
Alert: Bearish Key Level Break	on	a close below a level price was over	`Bearish Key Level Break (VAL 20105.75) on MNQ 5 @ 20088.25`
Alert: Bullish Key Level Rejection	on	a level held as support	`Bullish Key Level Rejection (High Vol 20110.25) on MNQ 5 @ 20136.00`
Alert: Bearish Key Level Rejection	on	a level held as resistance	`Bearish Key Level Rejection (Peak 20180.50) on MNQ 5 @ 20155.25`
Alert: Bullish Key Level Retest	on	a broken level retested from above	`Bullish Key Level Retest (Peak 20123.25) on MNQ 5 @ 20131.00`
Alert: Bearish Key Level Retest	on	a broken level retested from below	`Bearish Key Level Retest (Trough 20020.75) on MNQ 5 @ 20004.50`
Alert: Any Bullish Key Level Signal	off	any of the three bullish signals	`Any Bullish Key Level Signal (Peak 20123.25) on MNQ 5 @ 20130.50`

SWITCH	DEFAULT	FIRES WHEN	MESSAGE
Alert: Any Bearish Key Level Signal	off	any of the three bearish signals	`Any Bearish Key Level Signal (VAL 20105.75) on MNQ 5 @ 20088.25`

The message names the level or levels the signal fired on, then the instrument, the chart interval (`5`, `60`, `1D`, `30S`, `500T`) and the close of the bar at the instrument's tick size. The prices above are examples. The two **Any** switches are off because the six individual ones already cover every signal: turn on an **Any** switch instead of its three, not as well, or each signal arrives twice. **New Profile Key Level** is off because a rebuild can create a level every few bars. The signal filters (**Signal Levels**, **Signal Cooldown**, **Minimum RVOL**) apply to alerts too. The marker switches do not: **Signal Markers**, **Breaks**, **Rejections** and **Retests** off hide markers, and the alerts still fire.

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 `Bullish Key Level Break | Bullish Key Level Break (Peak 20123.25) on MNQ 5 @ 20130.50`. An e-mail has the subject `Supreme Dynamic Profile | Bullish Key Level Break` 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 | Supreme Dynamic Profile test on MNQ 5 @ 20130.50`. 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 committed 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 marker prints with no alert.

Using the values in a strategy or the Market Analyzer

Ten hidden series are exposed: `NewProfileKeyLevel`, `BullishBreak`, `BearishBreak`, `BullishRejection`, `BearishRejection`, `BullishRetest` and `BearishRetest` (1 on the closed bar of the event, 0 otherwise), and `DevelopingPoc`, `DevelopingVah` and `DevelopingVal` (a price under the Developing variants). Example: `if (SupremeDynamicProfile(...).BullishBreak[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 describes where volume traded on the bars of the window and how price behaved at those levels; it does not forecast where price goes next, and its percentages are not win probabilities.

04

Calculate mode and bar timing

- Default `Calculate.OnBarClose`: one engine commit per closed bar. The chart moves when a bar closes.
- `OnEachTick` / `OnPriceChange`: the closed bar is committed once, at the first tick of the next bar. The forming bar is previewed at most four times a second: the engine computes it and rolls everything back at once.
- A preview moves only what follows the newest price: the level colours and which levels are visible, the latest profile's position, the level labels and the table. It creates no level, builds no profile, fires no signal and raises no alert. The profile, the levels' life and every signal belong to closed bars, so all three modes give the same results bar for bar.
- The last historical bar, still forming when the chart loads, is previewed and is committed once it has closed.
- The profile is rebuilt on every **Refresh Every N Bars** closed bar, counted from the first bar the chart loaded, and only inside the newest **Calculation History** bars, counted back from the last closed bar when the chart loaded. Reloading with a different amount of history can therefore change which bars rebuild and how the first levels start. The sessions chapter has the detail.
- A marker on a closed bar does not change. An active level extends with the newest bar until it breaks, expires or is removed, and a stopped level is removed unless a history option keeps it. The profile, its labels and the table describe the newest bars and move with them, by design.
- **The final bar of a session is committed 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 a preview until the first tick after the reopen. Its alert is raised only if the reopen comes within two bar periods plus five seconds of that bar's close; otherwise the marker prints with no alert.
- Historical pass: one call per closed bar. The engine slot is the NinjaTrader bar index and the host walks it from the bar series, so the historical to realtime transition can neither double-feed nor skip a bar. 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.
- **Use Lower Timeframe** (off by default, second and minute charts only) adds one series of about the chart period divided by **Bar Granularity**. A closed chart bar reads the bars of that series stamped inside it, and a stamp past its close is never read, so it cannot look ahead. A forming bar reads none.
- A bar's OPEN time is its NinjaTrader close time minus the bar period, converted to UTC through the time zone chosen in NinjaTrader. Only the CSV dump reads it. Everything drawn is positioned by bar index.
- The price axis follows the bars, not the drawings (`IsAutoScale = false`), so a level far from price can lie outside the panel.

05

Outputs for strategies and the Market Analyzer

PLOT / SERIES	VALUES
`NewProfileKeyLevel` (plot 0, hidden)	1 on the closed bar where a profile rebuild created at least one level, 0 otherwise
`BullishBreak` (plot 1, hidden)	1 on the closed bar of an accepted bullish break, 0 otherwise
`BearishBreak` (plot 2, hidden)	1 on the closed bar of an accepted bearish break, 0 otherwise
`BullishRejection` (plot 3, hidden)	1 on the closed bar of an accepted bullish rejection, 0 otherwise
`BearishRejection` (plot 4, hidden)	1 on the closed bar of an accepted bearish rejection, 0 otherwise
`BullishRetest` (plot 5, hidden)	1 on the closed bar of an accepted bullish retest, 0 otherwise
`BearishRetest` (plot 6, hidden)	1 on the closed bar of an accepted bearish retest, 0 otherwise
`DevelopingPoc` (plot 7, hidden)	the POC when **POC Variant** is Developing and **Point of Control** is on; no value otherwise
`DevelopingVah` (plot 8, hidden)	the value-area high when **Value Area Variant** is Developing and **Value Area Levels** is on; no value otherwise
`DevelopingVal` (plot 9, hidden)	the value-area low under the same conditions; no value otherwise

NinjaTrader does not draw these plots. The chart shows the markers on the surface, and the developing paths as step lines on the surface, because NinjaTrader's plot styles have no step line. The series follow the accepted events and not the drawing switches, so **Signal Markers** off leaves them 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 seven events, and the developing series hold the latest profile's values.

Example in a strategy: `var sdp = SupremeDynamicProfile(...); if (sdp.BullishBreak[0] > 0) ...` (the accessor is generated by NinjaTrader on compile; every input except the three colours is one of its arguments). With **Use Lower Timeframe** on, a script that hosts the indicator must add the same series itself.

06

Parameters

The inputs in groups 01 to 11 keep the titles, order, defaults and ranges of the other editions (the group names carry a number prefix because NinjaTrader sorts groups alphabetically). Groups 12 Display, 13 Alerts and 14 NinjaTrader hold the platform settings.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
01 Components	Volume Profile	bool	true		`ShowProfile`	Shows the histogram of the latest profile.
01 Components	Point of Control	bool	false		`EnablePoc`	Shows the POC. Select its variant in 04 Key Level Variants. Turning POC off also disables fixed POC levels and their signals.
01 Components	Value Area Levels	bool	true		`EnableValueArea`	Enables VAH and VAL. Background shading is separate. Select Current, Historical, or Developing levels in 04 Key Level Variants.
01 Components	Background	bool	false		`ShowBackground`	Shades the latest profile. Select the background type under Background Variant.
01 Components	Volume Peaks	bool	true		`ShowPeaks`	Enables peak levels, with a Lines or Zones variant. Turning a level type off also disables its level signals.
01 Components	Volume Troughs	bool	false		`ShowTroughs`	Enables trough levels, with a Lines or Zones variant. Turning a level type off also disables its level signals.
01 Components	Ranked Volume Rows	bool	true		`ShowRanked`	Adds high- or low-volume rows that pass the filters.
01 Components	Historical Nodes	bool	false		`KeepHistory`	Keeps stopped node drawings. POC and value-area history use their own variants. History is off by default.
01 Components	Level Labels	bool	true		`ShowLevelLabels`	Shows spaced labels on visible active levels. Label content and text size are set in 08 Label Settings.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
01 Components	Metrics Table	bool	true		`ShowMetrics`	Shows the metrics table. Its layout is set in 09 Metrics Table.
01 Components	Visual Accents	bool	true		`ShowAccents`	Adds profile outlines and level highlights.
01 Components	Zone Center Lines	bool	false		`ShowCenterLine`	Adds dotted lines inside enabled zones.
01 Components	Emphasize Nearby Levels	bool	true		`FocusNearby`	Near-price emphasis changes drawing contrast only.
01 Components	Label Prices	bool	false		`ShowPrices`	Adds the exact price to each visible label.
01 Components	Signal Markers	bool	true		`ShowSignals`	Shows closed-bar signals for enabled fixed level types. This switch hides markers only; alerts still operate. Select the marker style under Marker Variant.
01 Components	Breaks	bool	true		`ShowBreaks`	Break events receive markers. This switch does not disable alerts.
01 Components	Rejections	bool	true		`ShowRejections`	Rejection events receive markers. This switch does not disable alerts.
01 Components	Retests	bool	true		`ShowRetests`	Retest events receive markers. This switch does not disable alerts. Retests require Stop Extending On Break to be disabled.
02 Level Management	Extend Previous Nodes	bool	true		`KeepOldActive`	Keeps old nodes active until break, expiry, or storage removal. The visible-node limit only hides extra drawings; those levels remain tracked. POC and value-area lines always follow the latest profile.
02 Level Management	Stop Extending On Break	bool	true		`StopOnBreak`	Ends a level after a close breaks through it. Disable it for later retests.
02 Level Management	Avoid Overlap	bool	true		`PreventOverlap`	Blocks new nodes that overlap active nodes or POC. Value-area lines are exempt.
02 Level Management	Level Expiry Bars	int	500	1 to 5000	`ExpiryBars`	A level stops this many bars after it was created.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
02 Level Management	Maximum Stored Levels	int	150	40 to 250	`MaxLevels`	Total active and stopped levels retained. Removes the oldest stopped level first, or the oldest active level when all are active.
03 Calculation	Lookback Mode	NinjaTrader.NinjaScript.SdpProfileMode	NinjaTrader.NinjaScript.SdpProfileMode.Fixed		`ProfileMode`	Fixed uses the selected lookback. Dynamic starts at the latest confirmed pivot, within the lookback limit.
03 Calculation	Lookback Limit	int	240	30 to 1500	`Lookback`	How many closed bars the profile covers at most.
03 Calculation	Pivot Period	int	35	2 to 250	`PivotLength`	Bars required on each side of a pivot. A new anchor is available only after the pivot is confirmed.
03 Calculation	Profile Resolution	int	64	20 to 100	`Resolution`	The number of rows of the profile.
03 Calculation	Refresh Every N Bars	int	5	1 to 100	`RefreshBars`	Rebuilds the profile on every Nth closed chart bar. Signals still use each chart-bar close. Set to 1 for each-bar profile updates.
03 Calculation	Value Area (%)	double	70.0	1.0 to 100.0	`ValueAreaPercent`	The share of the profile's volume the value area holds.
04 Key Level Variants	POC Variant	NinjaTrader.NinjaScript.SdpLevelVariant	NinjaTrader.NinjaScript.SdpLevelVariant.Current		`PocMode`	Current keeps the active POC line. Historical also keeps stopped POC lines. Developing shows the POC path without creating fixed POC levels or POC signals.
04 Key Level Variants	Value Area Variant	NinjaTrader.NinjaScript.SdpLevelVariant	NinjaTrader.NinjaScript.SdpLevelVariant.Current		`ValueAreaMode`	Controls VAH and VAL. Current keeps active lines. Historical also keeps stopped lines. Developing shows their paths without fixed value-area levels or their signals.
04 Key Level Variants	Peak Variant	NinjaTrader.NinjaScript.SdpNodeVariant	NinjaTrader.NinjaScript.SdpNodeVariant.Zones		`PeakMode`	Draws a peak as a line or as a zone.
04 Key Level Variants	Peak Max	int	2	1 to 8	`PeakCount`	Maximum local volume peaks selected per profile. Existing nodes can remain active when Extend Previous Nodes is enabled.
04 Key Level Variants	Trough Variant	NinjaTrader.NinjaScript.SdpNodeVariant	NinjaTrader.NinjaScript.SdpNodeVariant.Lines		`TroughMode`	Draws a trough as a line or as a zone.
04 Key Level Variants	Trough Max	int	1	1 to 8	`TroughCount`	Maximum local volume troughs selected per profile. Outer profile rows are excluded.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
04 Key Level Variants	Ranked Row Variant	NinjaTrader.NinjaScript.SdpNodeVariant	NinjaTrader.NinjaScript.SdpNodeVariant.Lines		`RankedMode`	Adds ranked volume rows even when they are not local peaks or troughs. Duplicate and overlapping rows can be skipped.
04 Key Level Variants	Highest Rows	int	1	0 to 5	`HighestRows`	Number of extra high-volume rows. Zero excludes that side. Rows must pass the volume and overlap filters.
04 Key Level Variants	Lowest Rows	int	0	0 to 5	`LowestRows`	Number of extra low-volume rows. Zero excludes that side. Rows must pass the volume and overlap filters.
04 Key Level Variants	Zone Radius (Rows)	int	1	0 to 5	`ZoneRadius`	Rows included on each side of a node. A radius of 1 creates a three-row zone. Lines mode uses one row center.
05 Profile Display	Placement	NinjaTrader.NinjaScript.SdpPlacement	NinjaTrader.NinjaScript.SdpPlacement.Right		`ProfilePlacement`	Right draws the histogram past the last bar. Left starts it at the first bar of the latest profile window.
05 Profile Display	Color Variant	NinjaTrader.NinjaScript.SdpProfileColor	NinjaTrader.NinjaScript.SdpProfileColor.Directional		`ProfileColorMode`	Directional colors each row by estimated up/down volume balance. Up / Down splits rows into two parts. Relative Volume uses one theme-aware color. Row width represents volume, not buying or selling pressure.
05 Profile Display	Width	int	60	5 to 200	`ProfileWidth`	Profile width in chart bars.
05 Profile Display	Offset	int	20	0 to 250	`ProfileOffset`	Right-side offset in chart bars. Left placement starts at the first bar of the latest profile window.
05 Profile Display	Background Variant	NinjaTrader.NinjaScript.SdpBackground	NinjaTrader.NinjaScript.SdpBackground.ValueArea		`BackgroundMode`	Shading for the latest profile only. Value Area shades the range between VAH and VAL.
06 Signal Settings	Marker Variant	NinjaTrader.NinjaScript.SdpMarkerVariant	NinjaTrader.NinjaScript.SdpMarkerVariant.Markers		`SignalDisplay`	Markers uses symbols only. Markers + Text also shows the event name. New levels cannot signal on their creation bar.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
06 Signal Settings	Signal Levels	NinjaTrader.NinjaScript.SdpSignalScope	NinjaTrader.NinjaScript.SdpSignalScope.Nodes		`SignalScope`	Nodes uses peaks, troughs, and ranked rows. All Levels also uses enabled fixed POC and value-area levels. Alerts can use tracked nodes hidden by the visible-node limit.
06 Signal Settings	Signal Cooldown	int	3	0 to 100	`SignalCooldown`	Minimum bar spacing between accepted signals on the same level. Also applies to alerts.
06 Signal Settings	Retest Window	int	30	1 to 500	`RetestWindow`	Maximum bars after a break for its first retest.
06 Signal Settings	Volume Average	int	20	1 to 200	`VolumeLength`	Bars in the average behind relative volume.
06 Signal Settings	Minimum RVOL	double	0.0	0.0 to 10.0	`MinimumRelativeVolume`	Signal volume divided by the preceding average volume. Zero disables the filter. Also applies to alerts.
06 Signal Settings	Signal Label Limit	int	80	10 to 300	`MaxSignalLabels`	Maximum stored signal markers. At most one label per direction per bar. Breaks take priority over retests, then rejections.
07 Appearance	Accent Variant	NinjaTrader.NinjaScript.SdpAccent	NinjaTrader.NinjaScript.SdpAccent.Signature		`VisualStyle`	Signature adds node-core shading, edge brackets, and line glow. Volume Balance replaces core shading with an up/down volume strip. Uncheck Visual Accents for plain drawings.
07 Appearance	Contrast	NinjaTrader.NinjaScript.SdpContrast	NinjaTrader.NinjaScript.SdpContrast.Bright		`ContrastMode`	Controls transparency. Custom reads the values in 11 Custom Transparency.
07 Appearance	Visible Active Nodes	int	8	3 to 100	`MaxVisibleNodes`	Limits active node drawings, not stored levels or alerts. Includes one nearest node from each available family, then fills remaining slots by distance to price. POC and value-area lines are separate. History has its own checkbox.
07 Appearance	Level Extension	int	5	0 to 100	`LevelExtension`	How many bars past the last bar an active level runs.
07 Appearance	Bullish Color	colour	#00FFBB		`BullishBrush`	The bullish color.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
07 Appearance	Bearish Color	colour	#FF1100		`BearishBrush`	The bearish color.
07 Appearance	Neutral Color	colour	#787B86		`NeutralBrush`	The neutral color. POC uses the chart's text color.
08 Label Settings	Label Content	NinjaTrader.NinjaScript.SdpLabelContent	NinjaTrader.NinjaScript.SdpLabelContent.VolumeShare		`LabelMetric`	Volume Share compares the row or zone with total profile volume. Row Strength compares the node row with the POC row. Up Volume uses candle direction, not actual buying volume. Values are fixed at creation and are not win probabilities.
08 Label Settings	Label Text Size	NinjaTrader.NinjaScript.SdpSize	NinjaTrader.NinjaScript.SdpSize.Tiny		`LabelSizeInput`	The text size of the level labels.
08 Label Settings	Label Max	int	8	1 to 18	`MaxLevelLabels`	Maximum labels on visible active levels. Labels use one right-side column. Labels too close in price are skipped without moving level prices.
09 Metrics Table	Table Variant	NinjaTrader.NinjaScript.SdpTableVariant	NinjaTrader.NinjaScript.SdpTableVariant.Compact		`TableMode`	Compact shows value-area position, POC distance, nearby visible levels, and relative volume. Detailed adds profile data, volume balance, level counts, and the last signal.
09 Metrics Table	Table Text Size	NinjaTrader.NinjaScript.SdpSize	NinjaTrader.NinjaScript.SdpSize.Small		`TableSizeInput`	The text size of the table.
09 Metrics Table	Position	NinjaTrader.NinjaScript.SdpTablePosition	NinjaTrader.NinjaScript.SdpTablePosition.TopRight		`TablePosition`	Where the table sits on the chart.
10 Advanced Calculation	Peak Detection (%)	double	7.0	1.0 to 50.0	`PeakDetection`	Neighbor rows checked on each side as a percentage of profile resolution. Larger values select broader peaks.
10 Advanced Calculation	Trough Detection (%)	double	7.0	1.0 to 50.0	`TroughDetection`	Neighbor rows checked on each side as a percentage of profile resolution. Larger values select broader troughs.
10 Advanced Calculation	Minimum Node Volume (%)	double	1.0	0.0 to 100.0	`MinimumNodeVolume`	Minimum node-row volume relative to the POC row. Empty rows are excluded.
10 Advanced Calculation	Node Smoothing Radius	int	1	0 to 5	`Smoothing`	Smooths row volume for local peak and trough detection only. Set to 0 to disable.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
10 Advanced Calculation	Node Match Tolerance (Rows)	double	0.75	0.0 to 5.0	`MatchTolerance`	Keeps an existing node when a new node of the same type is this close. Its price, bounds, and creation statistics stay fixed.
10 Advanced Calculation	Calculation History	int	2000	100 to 10000	`CalculationHistory`	Recent chart bars used for historical profile updates and signals. Earlier bars can supply lookback data. A moving start window can change initial level state after reload.
10 Advanced Calculation	Use Lower Timeframe	bool	false		`UseLowerTimeframe`	Uses lower-timeframe candles when available; otherwise chart candles. Volume is spread uniformly across each candle range. This is estimated volume at price, not exchange bid/ask data. Second and minute charts only.
10 Advanced Calculation	Bar Granularity	int	5	2 to 20	`Intragran`	The lower timeframe is approximately the chart timeframe divided by this number.
10 Advanced Calculation	Seconds	bool	false		`AllowSeconds`	Allows a lower timeframe below one minute. Seconds data is optional. The Detailed table reports lower-timeframe coverage.
11 Custom Transparency	Zone Fill	int	72	0 to 100	`ZoneFillTransparency`	Custom zone fill transparency. Lower values make zones more visible. Near-price emphasis can reduce these values further.
11 Custom Transparency	Border	int	48	0 to 100	`ZoneBorderTransparency`	Custom zone border transparency. Lower values make zones more visible. Near-price emphasis can reduce these values further.
11 Custom Transparency	History	int	78	0 to 100	`InactiveTransparency`	Custom history transparency.
11 Custom Transparency	Profile	int	15	0 to 100	`ProfileTransparency`	Custom profile base transparency. Profile rows with less volume use a softer setting.
12 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. Off: every drawing stays.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
12 Display	Pin Profile Inside Chart	bool	true		`PinProfileInside`	The profile is drawn 20 to 80 bars right of the last candle at the default offset and width. On: while the last candle is on screen, the profile and the level labels are moved left just enough to stay inside the chart. Off: they stay at their designed position past the last candle (add room with chart Properties > Right side margin, or scroll, to see them).
13 Alerts	Alert: New Profile Key Level	bool	false		`AlertNewLevel`	A profile rebuild creates at least one new level. Every alert appears in the Alerts Log window.
13 Alerts	Alert: Bullish Key Level Break	bool	true		`AlertBullBreak`	A close above a level price was under.
13 Alerts	Alert: Bearish Key Level Break	bool	true		`AlertBearBreak`	A close below a level price was over.
13 Alerts	Alert: Bullish Key Level Rejection	bool	true		`AlertBullRejection`	A level held as support.
13 Alerts	Alert: Bearish Key Level Rejection	bool	true		`AlertBearRejection`	A level held as resistance.
13 Alerts	Alert: Bullish Key Level Retest	bool	true		`AlertBullRetest`	A broken level retested from above. Needs Stop Extending On Break off.
13 Alerts	Alert: Bearish Key Level Retest	bool	true		`AlertBearRetest`	A broken level retested from below. Needs Stop Extending On Break off.
13 Alerts	Alert: Any Bullish Key Level Signal	bool	false		`AlertAnyBull`	Any of the three bullish signals. Use it instead of those three, not as well.
13 Alerts	Alert: Any Bearish Key Level Signal	bool	false		`AlertAnyBear`	Any of the three bearish signals. Use it instead of those three, not as well.
13 Alerts	Alert Log Line	bool	true		`AlertLogLine`	Writes each alert to the NinjaScript Output window.
13 Alerts	Alert Sound	bool	false		`AlertSound`	Plays a sound with each alert.
13 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.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
13 Alerts	Alert E-mail	bool	false		`AlertEmail`	Sends each alert by e-mail through the Mail share service configured in NinjaTrader (Tools > Options > Share).
13 Alerts	E-mail To	string	""		`EmailTo`	Recipient address for alert e-mails. Blank sends none.
13 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.
14 NinjaTrader	Debug Mode	bool	false		`DebugMode`	Writes diagnostics to sdp_nt_debug.log in the NinjaTrader 8 folder.
14 NinjaTrader	Dump Bars + Outputs CSV	bool	false		`DebugDumpCsv`	Writes every committed bar with the engine outputs to sdp_dump_<instrument> *<period>.csv, and with Use Lower Timeframe on each bar's intrabars to* *sdp_ltf*<instrument>_<period>.csv (parity checks).

96 parameters.

07

Sessions, time and volume

Sessions

The indicator has no session logic of its own: the profile's window is a count of closed bars, and the chart's Trading Hours template decides which bars exist. With an ETH template (for example `CME US Index Futures ETH`) the overnight bars count towards the window and bring their volume; with an RTH template they do not. Both are valid, and the profile differs, as it does for any volume profile. Nothing resets at a session start: a profile can span a session gap.

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 a bar's events are decided only once it has closed, so on a closed market the last bar stays a preview until the first tick after the reopen. Its alert is raised only if the reopen comes within two bar periods plus five seconds of that bar's close; otherwise the marker 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 indicator does not extend the chart's data, so the Data Series setting **Days to load** decides how many bars there are. Three things read that number:

- **Calculation History** (2,000 bars): a profile is built only inside this many bars back from the last closed bar when the chart loaded. Older bars still feed the lookback, the ATR and the volume average. The levels on the left of that point do not exist.
- **Refresh Every N Bars** (5): a profile is rebuilt on every Nth bar, counted from the first bar the chart loaded. Another amount of history shifts which bars rebuild.
- Levels live on from one profile to the next, so the levels on screen depend on where the calculation started.

So two charts that loaded different amounts of history can differ in their levels and signals for a while, and a chart reloaded after a long session can differ from what it showed live. Compare charts with the same **Days to load** and Trading Hours template. With fewer closed bars than **Calculation History**, everything the chart holds is used.

With **Keep Chart Object Limits** on, the newest 500 boxes, 500 lines and 500 labels 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 time zone chosen under Tools > Options. The bar's open time is reconstructed as close time minus the bar period and converted to UTC through that zone; on day and longer bars from the stamp's date, on non-time bars from the previous bar's stamp. Only the CSV dump reads it. Everything drawn is positioned by bar index. Slots past the last bar, where the histogram and the label column sit, are placed at the average bar spacing of the view.

Volume

Volume on a futures chart is real contract volume, so the profile shows exchange volume by price row, as far as a bar can tell. The indicator works from bars, not from tick data: a bar's volume is spread evenly over the rows its range covers, and it is counted as up volume for a bar closing above its open, as down volume for a bar closing below it, and half and half for a bar closing at its open. So this is estimated volume at price and an estimated up and down split, not exchange bid and ask data. On a forex feed the volume is usually tick volume, and two feeds' profiles of one symbol differ.

Relative volume is the bar's volume divided by the average of the **Volume Average** bars before it. On a feed without volume no profile exists, the table reads `No volume profile`, and no levels appear.

Lower timeframe

Use Lower Timeframe (off by default) replaces each closed bar's single candle with smaller candles: about the chart period divided by **Bar Granularity**, rounded up to the next valid timeframe. With the default 5, a 5-minute chart reads 1-minute bars, a 15-minute chart 3-minute bars, a 60-minute chart 12-minute bars. Without **Seconds** the lower timeframe is never below one minute, so a 1-

minute chart stays with its chart candles.

- It adds one series of the chart's own instrument, loaded over the chart's own range. A long chart on a small timeframe loads more data.
- A chart bar reads the series' bars stamped inside it. A stamp past the chart bar's close is never read, so nothing can look ahead. A forming bar reads none.
- A chart bar with no lower-timeframe bars, for example older than that series' history, uses its own candle. The **Sampling** row of the Detailed table reports the share of the window that had them.
- Second bars are built by NinjaTrader from tick data, which not every connection delivers for history.
- Second and minute charts only. NinjaTrader stamps a day, week or month bar with a date and no time, and tick, volume and range bars have no period to divide, so those charts use the chart candles.

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, and on a roll day the merged series can carry only one contract's volume. A profile whose window spans a non-back-adjusted roll stretches over the gap, with thin rows inside it, and levels from before the roll can read as broken. 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. One data series; with **Use Lower Timeframe** on, one more of the same instrument.
Instruments	Any instrument with bars and volume: futures, micro futures, forex, stocks. Tick sizes such as 0.25 and 1/32 print in full.
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. **Use Lower Timeframe** works on second and minute charts only; on any other chart it uses the chart candles, and the Detailed table reads 0 % coverage.
Calculate modes	OnBarClose (default), OnEachTick, OnPriceChange. A bar is committed once, when it has closed, and all three modes commit the same results. OnEachTick and OnPriceChange also preview the forming bar at most four times a second. With them, 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 sessions chapter.
Multi-instance	Yes. Every instance paints its own surface, and every alert id carries the instrument, the period and an instance hash.
Market Analyzer	The ten series can be used as columns; there is no chart, so nothing is drawn. Not advertised as a scanner.
Strategies	The ten series can be referenced from a strategy. Every input except the three colours is a `[NinjaScriptProperty]`. With **Use Lower Timeframe** on, the hosting script must add the same series itself.
Data requirements	Bars with volume. On a feed without volume no profile exists: the table reads `No volume profile` and no levels appear.
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 POC, the history tints, the table and the label plates read the chart's own background and text colours when the indicator starts. After changing them, reload the chart (F5).
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.
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 table reads `No volume profile`	The bars of the window carry no volume on this feed. A volume profile needs volume.
The table reads `Waiting for closed bars` under Detailed	The first profile needs at least ten closed bars, inside **Calculation History**, on a bar where **Refresh Every N Bars** comes round. Load more bars or wait.
The profile is not next to the last candle	**Offset** and **Width** place it in bars past the last bar, and **Pin Profile Inside Chart** moves it left just enough to fit. With it off, add room with chart Properties > Right side margin. With the last bar scrolled off screen nothing is pinned, so the profile is out of view with it.
The profile covers the newest candles	That is the pin keeping it inside the chart. Add room with chart Properties > Right side margin, lower **Width**, or choose **Placement = Left**.
The profile changes on a closed bar and not on the next ones	It is rebuilt on every **Refresh Every N Bars** closed bar. Set 1 to rebuild on every bar.
No POC line	**Point of Control** is off by default. Developing shows a path and no fixed line.
Fewer levels than expected	**Visible Active Nodes** limits the drawings, **Avoid Overlap** blocks a new node that overlaps an active one, **Minimum Node Volume (%)** filters thin rows, and **Stop Extending On Break** ends a level at its break.
A level moved colour	A level turns green when a bar closes above it and red when a bar closes below it. Its strength follows the distance to the newest price. A level's price does not move.
No markers	**Signal Markers** and the **Breaks**, **Rejections**, **Retests** switches hide markers only. A signal needs a level that **Signal Levels** allows: at the default the POC and the value area give no signal. **Minimum RVOL** and **Signal Cooldown** also filter.
No retest markers	A retest needs **Stop Extending On Break** off, because at the default a level stops at its break.
No marker 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 no earlier signal	By design. Signals are decided on closed bars in every Calculate mode. OnEachTick only makes the colours, the labels and the table follow the forming bar.
The oldest markers are gone	**Signal Label Limit** keeps the newest 80 by default. **Keep Chart Object Limits** retires the oldest boxes, lines and labels past 500 each.
A label is missing on a visible level	**Label Max** limits the labels, nearest first, and a label too close in price to a nearer one is skipped. The level's price does not move.
A symbol on a label or a marker looks different from another PC	◆ ● ○ ─ ▲ ▼ △ ▽ are not in Trebuchet MS, so Windows supplies them from another font. Check that the Windows symbol fonts are installed.
A level lies outside the panel	The price axis follows the bars, not the drawings. Drag the price axis to see it.
The table covers a label	The table is painted on top. Choose another **Position** in **09 Metrics Table**.
Colours look wrong after changing the chart background	The indicator reads the chart's colours when it starts. Reload the chart (F5).
Use Lower Timeframe changes nothing	It works on second and minute charts only, and the lower timeframe must be below the chart's period: on a 1-minute chart it needs **Seconds** on. The **Sampling** row of the Detailed table shows the coverage.
Sampling reads a low coverage	The lower-timeframe series reaches less far back than the chart, or the connection delivers no second bars for history. Those bars use the chart candle.
The chart loads slowly with Use Lower Timeframe	The series is loaded over the chart's own range. Lower **Days to load**, or lower **Bar Granularity**: a smaller number gives a larger lower timeframe.
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 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. Retest alerts need **Stop Extending On Break** off.
Each signal arrives twice	An **Any** switch is on together with its individual switches. Use one or the other.

SYMPTOM	CAUSE AND FIX
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.
Values differ from another platform on the same symbol	Bars and volume differ by feed and session template, and a volume profile is built from them. Compare on identical bars (see the chapter on how this edition was verified).
The picture differs between two charts, or after a reload	The levels are built from the bars before them, and the refresh count starts at the first bar loaded. Compare charts with the same **Days to load** and Trading Hours template.
Text is tiny or huge	Sizes follow the Windows display scaling, **Label Text Size** and **Table Text Size**.
An error line names Supreme Dynamic Profile in the Log tab	Set **14 NinjaTrader > Debug Mode** on, reproduce it, and send `Documents\NinjaTrader 8\sdp_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 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 on EURUSD M5, M15 and H1 and XAUUSD M5, with 68 input sweeps on EURUSD M5, over 233 runs. Values were compared exactly, colours as ARGB, drawn objects field by field.

GATE	COMPARED	MISMATCHES
every calculated value, level and profile row, every bar	179,335,885 cells	0
the step plots' colours	10,223,865 cells	0
drawn objects (boxes, lines, labels, polylines, table cells)	155,487 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 pivot, ATR and volume series, the profile with 64 of its rows, the level checksums and the newest and oldest level, the signal priorities, the object counts, the developing values and the nine 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 1 Minute, historical load, NinjaTrader 8.1.9.0, default inputs	4,923	221	1,087,983	0
MNQ 1 Minute, across the switch from history to live bars, Calculate = OnEachTick	4,191	221	926,211	0

Bar times matched on every row, and the replay reads its prices from the same file. On the historical run a bullish break fired on 33 bars, a bearish break on 28, a bullish rejection on 74 and a bearish rejection on 58, the same bars on both sides. No retest fired: at the default inputs a level stops at its break.

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: **Use Lower Timeframe** was off in this run; 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 **14 NinjaTrader > Debug Mode** on, reproduce it once and attach `sdp_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-dynamic-profile/. 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 Dynamic Profile is a technical analysis tool. It draws a volume profile computed from historical bars, key levels taken from that profile, and markers where a closed bar broke, was rejected by or retested a level. It does not place, modify or close orders and it gives no trade advice. The volume at price and the up and down split are estimates made from bars, not exchange bid and ask data. The percentages on the labels describe the profile when a level was created; they are not win 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.